



**Universidad  
Politécnica  
de Cartagena**



**industriales**  
etsii UPCT

# **Aplicación docente de sistema SCADA y sensor inteligente sobre maqueta de pruebas**

**Titulación:** Ingeniero Técnico Industrial  
esp. Electrónica Industrial  
**Intensificación:** Automática  
**Alumno/a:** Miguel Ángel López  
**Director/a/s:** Fco. José Ortiz Zaragoza

Cartagena, 9 de octubre de 2014

# Agradecimientos

A mi familia por todo lo que me ha aportado y enseñado en todo este tiempo y a Francisco José Ortiz Zaragoza, director del proyecto, por su paciencia infinita, apoyo, comprensión y ayuda durante la elaboración de este trabajo.



# Tabla de contenidos

Capítulo 1. Introducción al proyecto y objetivos	pág. 1
1.1. Introducción.	pág. 1
1.2. Objetivos.	pág. 2
1.3. El sistema físico.	pág. 2
1.4. Estructura del documento.	pág. 4
Capítulo 2. Evolución de los sistemas de control y visualización. Los sistemas SCADA.	pág. 5
2.1. Un poco de historia.	pág. 5
2.2. Sistemas de visualización.	pág. 7
2.3. Interfaz Hombre-Máquina (HMI).	pág. 10
2.4. Los sistemas SCADA .	pág. 11
2.4.1. Introducción y contexto.	pág. 11
2.4.2. Descripción de un sistema SCADA.	pág. 11
2.4.3. Objetivos de un SCADA.	pág. 12
2.4.4. Características de los sistemas SCADA.	pág. 13
2.4.5. Prestaciones.	pág. 14
2.4.6. Requisitos.	pág. 14
2.4.7. Ventajas.	pág. 15
2.4.8. Estructura hardware de los sistemas SCADA.	pág. 16
2.4.9. ¿Cuándo implementar un sistema SCADA?	pág. 19
2.4.10. Implementando un SCADA.	pág. 20
2.4.11. El software SCADA.	pág. 20
2.4.12. Estructura del software SCADA.	pág. 22
2.4.13. Comunicaciones.	pág. 26
2.4.14. Comunicaciones entre aplicaciones.	pág. 27
2.4.15. Evolución del software.	pág. 30
2.4.16. Comparativas entre software SCADA.	pág. 31
2.5. Otras filosofías de control SCADA y PLC frente a DCS.	pág. 33
2.6. El SCADA de Siemens.	pág. 35
Capítulo 3. La maqueta.	pág. 37
3.1. La maqueta.	pág. 38
3.1.1. Estructura metálica.	pág. 39
3.1.2. Instalación hidráulica.	pág. 40
3.2. Sensorización de la maqueta.	pág. 47
3.2.1. Sensores de salida digital.	pág. 48
3.2.2. Sensores de salida analógica.	pág. 50
3.3. Instalación eléctrica de la maqueta.	pág. 63

3.3.1. El cuadro eléctrico.	pág. 63
3.3.2. Esquemas eléctricos.	pág. 66
3.4. El autómata de la maqueta. Unidad de Nais FPΣ C32T2.	pág. 68
3.4.1. Entradas y salidas del autómata.	pág. 68
3.4.2. Módulos de entradas analógicas.	pág. 69
3.5. Modificaciones de la maqueta. Banco de pruebas Siemens.	pág. 70
3.6. El autómata de Siemens.	pág. 71
3.6.1. Fuente de alimentación.	pág. 73
3.6.2. Módulos de E/S.	pág. 75
3.7. Duplicado Entradas/Salidas del cuadro original.	pág. 79
3.7.1. Duplicado E/S.	pág. 80
3.8. Sensor inteligente SITRANS Probe LU.	pág. 83
3.8.1. Pasarela PROFIBUS DP/PA.	pág. 84
Capítulo 4. Conexiones y Software.	pág. 86
4.1. Introducción.	pág. 86
4.2. Asociación de Entradas y Salidas del sistema.	pág. 88
4.2.1. Entradas y salidas digitales.	pág. 88
4.2.2. Entradas y salidas analógicas.	pág. 90
4.3. Conexionado de Entradas y Salidas.	pág. 91
4.4. Conexión y comunicación con SITRANS Probe LU.	pág. 94
4.4.1. PROFIBUS.	pág. 94
4.5. Software necesario.	pág. 97
4.5.1. Versiones instaladas.	pág. 97
4.5.2. Integración de programas.	pág. 99
Capítulo 5. SITRANS Probe LU.	pág. 100
5.1. Introducción a los sensores inteligentes.	pág. 100

5.2. Evolución de los sensores inteligentes.	pág. 103
5.3. Arquitectura general de un sensor inteligente.	pág. 105
5.3.1. Procesado de señal.	pág. 105
5.3.2. Control digital.	pág. 106
5.3.3. Comunicación.	pág. 107
5.4. Sistemas de comunicación para sensores.	pág. 108
5.4.1. Protocolo HART.	pág. 109
5.4.2. El bus de campo.	pág. 111
5.5. SITRANS Probe LU.	pág. 115
5.5.1. Información general.	pág. 115
5.5.2. Instalación y calibración.	pág. 118
5.5.3. Funcionamiento.	pág. 122
5.5.4. Comunicación.	pág. 124
5.5.5. Software y configuración del sensor.	pág. 127
Capítulo 6. Programación del PLC.	pág. 128
6.1. El entorno de Step 7.	pág. 128
6.1.1. Interfaz de configuración hardware.	pág. 129
6.1.2. Interfaz de programación (OB1).	pág. 130
6.2. Configuración hardware.	pág. 131
6.3. Ajustar la interfaz PG/PC.	pág. 142
6.4. Asociación de variables.	pág. 145
6.5. Aplicaciones básicas. Comprobar E/S y lectura del sensor.	pág. 147
6.6. Guardar y cargar el programa.	pág. 152
6.7. Programa todo/nada con E/S digitales.	pág. 155
6.8. Lectura y escritura de entradas y salidas analógicas.	pág. 161
6.8.1. Lectura de una entrada analógica.	pág. 162
6.8.2. Escritura de una salida analógica.	pág. 165
Capítulo 7. Programación SCADA.	pág. 166

7.1. El entorno de WinCC flexible.	pág. 166
7.1.1. Adecuar la interfaz.	pág. 167
7.2. Configurar la conexión en WinCC flexible.	pág. 169
7.2.1. Configuración Hardware.	pág. 169
7.3. Asociación de variables.	pág. 174
7.4. Crear las interfaces del SCADA.	pág. 176
7.4.1. Cargar librerías.	pág. 176
7.4.2. Interfaz central del proceso.	pág. 178
7.5. Lectura del sensor inteligente SITRANS Probe LU en WinCC flexible.	pág. 181
7.6. Navegación de imágenes y compilación del trabajo.	pág. 182
Capítulo 8. Conclusiones y trabajos futuros.	pág. 184

# Capítulo 1:

## Introducción al proyecto y objetivos.

---

En este capítulo de introducción al proyecto, se dará a conocer el sistema físico, su funcionamiento y las partes o elementos importantes que lo componen. Además, también se explicarán los objetivos perseguidos en este trabajo y la motivación por la que se ha elaborado.

### 1.1. Introducción al proyecto.

Con las primeras industrias mecanizadas aparecidas a finales del siglo XIX, posibilidades éstas gracias a la Revolución Industrial, y su eclosión en las primeras décadas del siglo XX, se hizo necesario tener bajo control cada uno de los procesos que componían tales industrias. Pero es sobre todo a partir de la segunda mitad de dicho siglo cuando se manifiesta de forma más notoria la necesidad de supervisar, controlar y actuar directamente, si fuera necesario, sobre estos procesos.

Las grandes compañías crecen, aumentando y diversificando su producción y servicios. Para abastecer un mercado y una demanda en auge se construyen plantas de producción enormes. Cada planta necesita un sistema de control para poder obtener la información necesaria sobre la producción y tomar las decisiones pertinentes. Estos controles se realizaban por medio de operarios especializados. Con el paso del tiempo, se hizo patente la necesidad de obtener una información más precisa y detallada de los procesos, así como actuar sobre ellos de forma directa y con menos mano de obra, mejorando así la producción. La automatización, tal y como se entiende en nuestros días, ha visto la luz y hay que controlar que todo funcione correctamente.

Hoy día, gracias a los avances de la electrónica (microprocesadores, sensores, autómatas...) un solo operario puede supervisar, controlar y modificar cualquier proceso automatizado de producción desde una estación de trabajo o de control. Las ventajas de tales estaciones de trabajo son:

- Sensorización de cada proceso.
- Monitorización de todos los procesos.
- Fácil instalación.
- Rápida adquisición de datos.
- Interactuación operario-proceso más efectiva.

Partiendo de una maqueta de otro Proyecto Fin de Carrera (*PFC* en lo sucesivo) cuyo propósito era simular el llenado y vaciado de los depósitos que la componían, se tomó la decisión de implementar en dicha maqueta un sistema de control completo. La maqueta consta de tres depósitos de metacrilato con sensores de medición de nivel cada uno y electro-válvulas para el llenado y el vaciado de los mismos. Con todo esto, surge la motivación de crear un sistema de control del proceso, de la necesidad de ofrecer un ejemplo claro y palpable al alumnado de dichos sistemas.

Por otro lado, cabe destacar la incorporación en este *PFC* del sensor inteligente *Sitrans Probe LU 6 m.* de la marca *Siemens*, siendo su instalación, integración con el sistema de control y configuración uno de los puntos principales del proyecto, cuyos objetivos se enumerarán en el siguiente punto.

## 1.2. Objetivos

En este apartado se van a enumerar los objetivos y se hará una breve descripción de cada uno de ellos. Los objetivos de este proyecto son los siguientes:

- Instalación hardware y software necesario para el sistema de control.
- Parametrización del sensor inteligente *SITRANS Probe LU 6m* y su correcto manejo.
- Visualización de todo el proceso desde una estación de trabajo mediante una interfaz de Supervisión, Control y Adquisición de Datos (*SCADA* en lo sucesivo).
- Crear un sistema de control a pequeña escala para que el alumno tengan un ejemplo palpable y veraz de dichos sistemas, siendo capaz además, de su manipulación, sirviéndole de guía docente en un futuro.
- Configuración de ejercicios prácticos para el alumnado.

## 1.3. El sistema físico

Vamos a explicar de forma resumida, a modo de introducción, en qué consiste nuestro sistema físico y las partes que lo componen. En los sucesivos capítulos del proyecto profundizaremos más en los elementos que componen abordando cada una de sus partes con más detalle. Dicho sistema se puede dividir en tres partes:

1. La maqueta. Compuesta por los tres depósitos y sensorizados cada uno de ellos. Como ya se ha comentado con anterioridad, se le ha acoplado el sensor inteligente de *Siemens*.
2. El autómatas o Controlador Lógico Programable (*PLC* de sus siglas en inglés y en lo sucesivo) que se encargará de activar o desactivar distintos componentes de la maqueta.
3. La estación de trabajo o puesto de control desde donde se visualizará y controlará todo el proceso. Está compuesto por un ordenador o *PC* con una tarjeta de comunicación y los correspondientes programas informáticos.

Las distintas partes de nuestro sistema físico se comunican entre sí mediante el bus de campo *PROFIBUS*. No hay que pasar por alto en esta introducción, que si bien el bus de campo es *PROFIBUS*, se utilizan dos estándares distintos. Para comunicar la estación de trabajo con el *PLC*, se utiliza el estándar *PROFIBUS DP*, mientras que para comunicar al *PLC* con el sensor *Sitrans Probe LU 6m.* se utiliza una pasarela *PROFIBUS DP/PA* ya que dicho sensor sólo se

comunica mediante *PROFIBUS PA*. La *Figura 1* muestra el sistema físico completo mientras que en la *Figura 2* podemos ver su organización en bloques.



Figura 1: Sistema físico

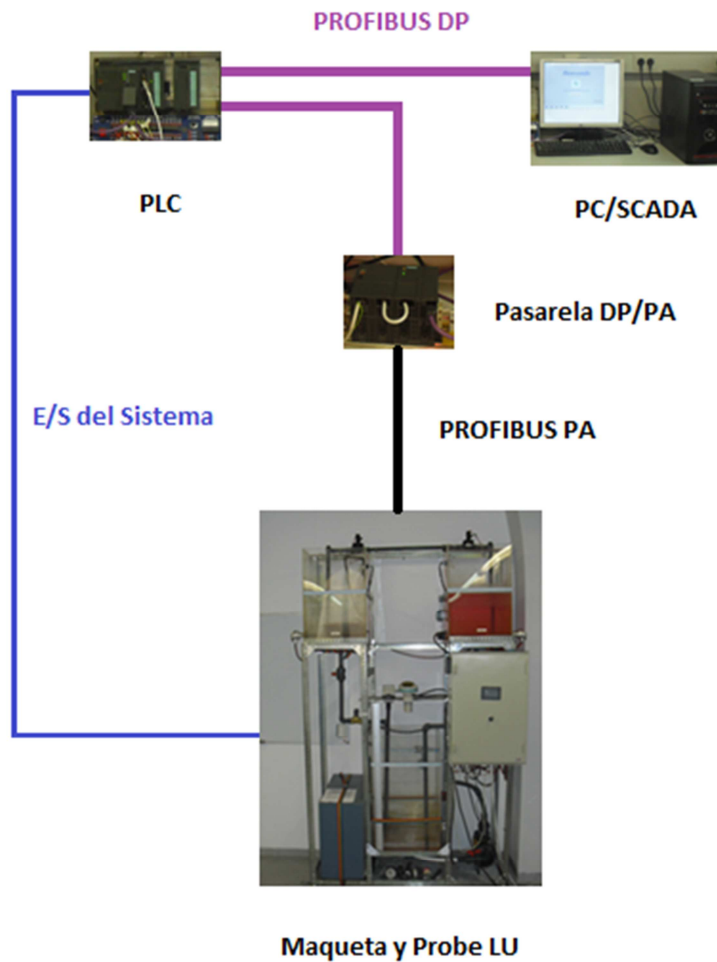


Figura 2: Organización por bloques

## 1.4. Estructura del documento

Además de esta introducción o primer capítulo, este documento consta de ocho capítulos más, haciendo un total de nueve, incluyendo varios anexos e información necesaria, amén de la bibliografía. El orden de capítulos queda como sigue:

- **Capítulo 2: Evolución de los sistemas de control y visualización. Los sistemas SCADA.** Breve reseña histórica de los sistemas de control hasta llegar al SCADA.
- **Capítulo 3: La maqueta.** Donde se explican todos los elementos que la conforman y las modificaciones pertinentes para este proyecto.
- **Capítulo 4: Conexiones y software.** Donde se explica las conexiones realizadas, así como el software utilizado para la elaboración del sistema de control.
- **Capítulo 5: SITRANS Probe LU.** Donde se hace una especial reseña al sensor inteligente de Siemens.
- **Capítulo 6: Programación del autómatas.** Donde se explica y muestra la programación utilizada en el PLC para el sistema de control
- **Capítulo 7: Programación del sistema de control, SCADA.** Como se han programado y configurado las distintas interfaces que componen el sistema de control.
- **Capítulo 8: Conclusiones y trabajos futuros.** Donde se exponen las distintas conclusiones a las que se ha llegado en este PFC y sus aplicaciones y trabajos futuros.
- **Anexos.** Información adicional de todo el trabajo realizado.



## Capítulo 2:

# Evolución de los sistemas de control y visualización. Los sistemas SCADA.

---

En el segundo capítulo se hará un recorrido histórico de los sistemas de control hasta llegar los sistemas de Supervisión, Control y Adquisición de Datos (SCADA en lo sucesivo) y su aplicación al proyecto que nos ocupa.

### 2.1. Un poco de historia.

En un principio, todos los sistemas de control tenían un funcionamiento manual, por lo que se requería de la presencia continua de operarios, que debían leer los datos de los distintos indicadores y calibrar los actuadores según la necesidad del proceso en cuestión. Estos operarios debían circular por cada planta, nave industrial o cadena de montaje cada cierto tiempo para comprobar que todo funcionara correctamente.

Los primeros sistemas de control eran puros paneles repletos de relojes de medida de señales de temperatura, pilotos de estado, etc. centralizados con capacidad sólo de captar información y de dar orden de marcha a alguno de los equipos de control.

Las primeras mejoras se realizaron en los sistemas de visualización, los paneles de control se hacían más pequeños gracias a la reducción en el número de relojes y de pilotos mediante la existencia de conmutadores que permitían cambiar las variables visualizadas.

La evolución de los sistemas viene marcada claramente por la evolución de la electrónica y de la informática. A finales de la década de los 50, el uso del transistor en los ordenadores marcó el inicio de equipos más pequeños, rápidos y versátiles de lo que permitían las máquinas con válvulas. Debido a que estos tenían un consumo mucho menor de energía, una vida útil más prolongada, y un tamaño menor, todos los equipos se hicieron más pequeños y su fabricación se hizo más barata.

Diez años más tarde, a finales de la década de los 60, apareció el circuito integrado (CI), que posibilitó la fabricación de varios transistores en una única pastilla de silicio. Los sistemas de control eran grandes armarios llenos de tarjetas electrónicas que centralizaban toda la información.

El circuito integrado permitió reducir más el precio, el tamaño y los porcentajes de error. A mediados de la década de los 70, el microprocesador se convirtió en una realidad. Con la llegada del microprocesador aparecieron los primeros sistemas con capacidad de realizar control digital directo (DDC). Eran sistemas que incorporaban la capacidad de captar señales y de tomar decisiones en base a parámetros prefijados.

A principios de la década de los 80, tras la aparición del ordenador personal, el uso de los microprocesadores se extendió y su coste se redujo considerablemente haciendo viable la aparición de controladores micro-procesados pequeños que permitieron el DDC descentralizado.

Todos estos pequeños micro-controladores tenían la posibilidad de comunicar sus datos a través de las líneas de comunicación (*BUS*) que los unían entre sí. Con cada uno de los nuevos micro-controladores, aparecía un nuevo *BUS* de comunicación. Los protocolos (lenguaje empleado por los micro-controladores) de comunicación proliferaban, cada uno de ellos sólo era válido para un tipo de micro-controlador.

A principios de los 90, aparecen ya los primeros sistemas de control con topología de *BUS* único, compatibles con micro-controladores de distinto propósito, comienzan a estandarizarse protocolos como *PCL5* de *Alan-Bradley*, *Optomux Opt022*, y otros que, por su diseño, los hacen idóneos para el intercambio de información de diversos formatos.

Las topologías de los sistemas de control comienzan a simplificarse, al bus de supervisión de micro-controladores distribuidos se le incorpora un *BUS* o *LAN* de gestión de datos en el que poco a poco se van usando tecnologías de red existentes como *ArcNet* o posteriormente *Ethernet*.

Los sistemas comienzan a evolucionar de forma similar a la que lo hacen los ordenadores personales, ya no es sólo importante que el sistema controle bien los equipos, se hace imprescindible que sea capaz de mostrar de forma sencilla los ciento o miles de datos que maneja. Con la aparición de los sistemas operativos gráficos, el puesto de trabajo se convierte en algo fácil de manejar por el operario.

Las actuales tendencias del mercado de sistemas dirigen los desarrollos de los sistemas de control en:

- Creación de programas para el puesto de gestión compatibles con los estándares de sistemas operativos.
- Empleo de buses con protocolos de comunicación de propósito general. Comienzan a extenderse el uso de protocolos como *Echelon Lon \Vorks*, *EIE* o *PROFIBUS*, cuya finalidad teórica es la posibilidad de conectar micro-controladores de distintos fabricantes.
- Integración de equipos de terceros.
- Intercambio de información con otros paquetes de software, datos en tiempo real como temperaturas, datos históricos acumulados, horarios. El uso de tecnologías para el intercambio dinámico de datos entre programas como *DDE (MS Windows)*, la selección de datos como *ODBC*, *SQL*, o el control de las aplicaciones de control desde terceras aplicaciones vía *OLE*, toman fuerza en los programas para los puestos de gestión.

Estas últimas tecnologías son estándares en la inmensa mayoría de los programas de aplicación, como hojas de cálculo, editores de texto, gestores de bases de datos, y permiten la generación de informes, gráficos, etc. totalmente adaptados a cualquier necesidad particular de forma automatizada, algo que hace muy poco tiempo resultaba imposible.

En la *Figura 1* puede verse un ejemplo de un sistema de control teniendo en cuenta lo explicado anteriormente, aplicado a la domótica en una vivienda:

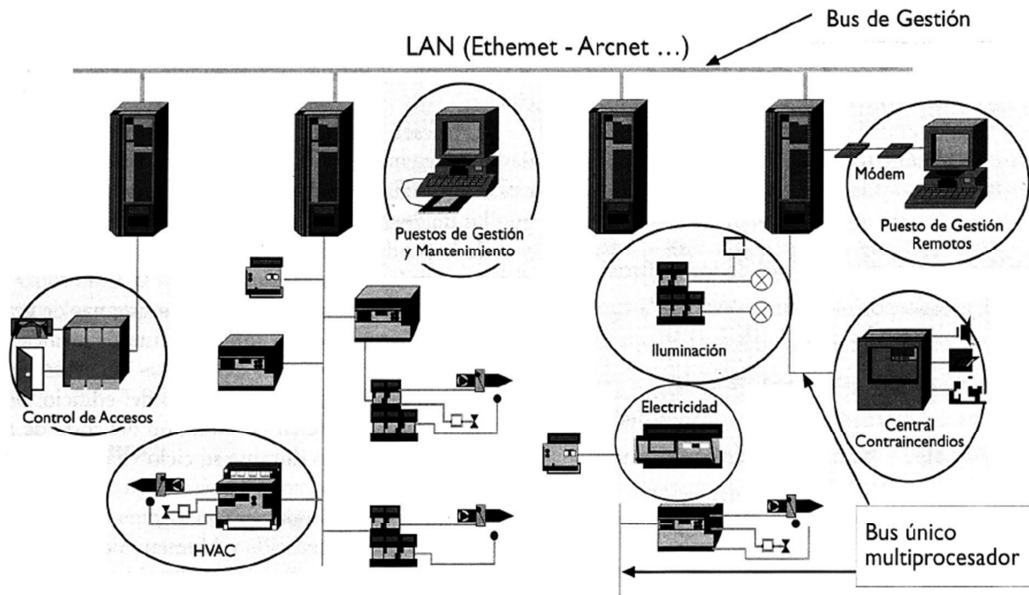


Figura 1: Ejemplo de un sistema de control aplicado a la domótica

Si se observa la imagen anterior, podemos observar como un bus general (de gestión) comunica a los distintos autómatas con el puesto de gestión general. Estos autómatas a su vez están conectados a distintos sensores y actuadores que se encargan de recopilar información sobre el estado de la vivienda y actuar en consecuencia. También se observa que se puede controlar todo el sistema de forma remota mediante un módem conectado a otro puesto de control. Esta es la configuración básica de todos los sistemas de control.

Por tanto nos encontramos en un momento en el que las antiguas culturas de ocultismo del conocimiento (*Know-How*), van dando paso a sistemas abiertos, capaces de comunicarse con terceros, transmitir datos a sistemas ofimáticos y dispuestos para ser controlados o supervisados a su vez por otros sistemas. Sistemas de Control, cuyo objetivo final es el de facilitar el uso y la explotación del sistema, suministrando, al gestor del mismo, la información precisa en formatos adaptables a la ofimática actual.

## 2.2. Sistemas de visualización.

Como ya se ha comentado anteriormente, gracias a los sistemas operativos gráficos, se abrió una nueva etapa a la hora de representar el estado de cualquier proceso o instalación controlada por automatismos. Hoy día, para representar cualquier proceso y su sistema de control disponemos de varias herramientas o sistemas de visualización. Pero antes vamos a hacer un pequeño repaso desde el origen más cercano hasta nuestros días.

Alrededor de los años sesenta la tendencia en automatización era la de que cada fabricante debía resolver sus problemas de control por sí solo. Quien se encontraba ante un problema de automatización desarrollaba un elemento electrónico específico para solventarlo. Una memoria reducida era lo normal en estos elementos, por lo cual necesitaban comunicarse constantemente con sus sistemas de control centrales para enviar los datos. Incluían una serie de entradas y salidas fijas y utilizaban generalmente lenguajes de programación poco conocidos.

Los años setenta ven aparecer una nueva generación de autómatas de la mano de fabricantes de equipos eléctricos como *Siemens*, *Square-D*, o *Allen-Bradley*. Implementaron autómatas capaces de controlar grandes cantidades de entradas y salidas, ideales para industrias tales como la automoción. No se trataba de entornos amigables, por lo cual estos controles estaban

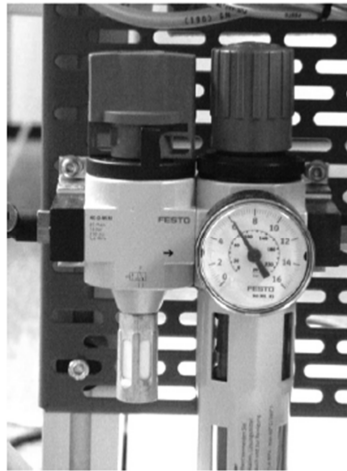
diseñados para soportar las condiciones más severas y, por tanto, eran grandes, pesados y muy caros.

Otra de las consecuencias de la evolución de la electrónica, que ya se ha comentado en el punto anterior y que significó un gran avance en estos sistemas también, fue la reducción de los componentes, lo que permitió realizar una disminución progresiva de tamaño, peso y coste en todos los niveles industriales de control.

Resultado de esto fue la introducción de los micro-PLC, en los años ochenta. Permitían realizar controles modulares que se adaptaban a las necesidades del momento y venían provistos ahora de sistemas de programación genéricos (ladder o escalera), lo que les deparó un éxito inmediato en todo el ámbito industrial.

Por otro lado, de una forma u otra, cada vez que se ha realizado el control de un sistema, grande o pequeño, ha sido necesario tener información visual de cómo está funcionando. Así, a medida que los sistemas de control han ido evolucionando y se han hecho cada vez más complejos, ha aumentado también la complejidad de los elementos que proporcionan la información al usuario.

De un simple indicador de aguja, que representa una variable del proceso (por ejemplo, la presión de aire en una instalación neumática), se ha llegado a grandes paneles sinópticos que muestran el estado de grandes instalaciones (por ejemplo, una refinería) tal y como puede verse en las *Figuras 2 y 3*.



*Figura 2: Indicador de presión*

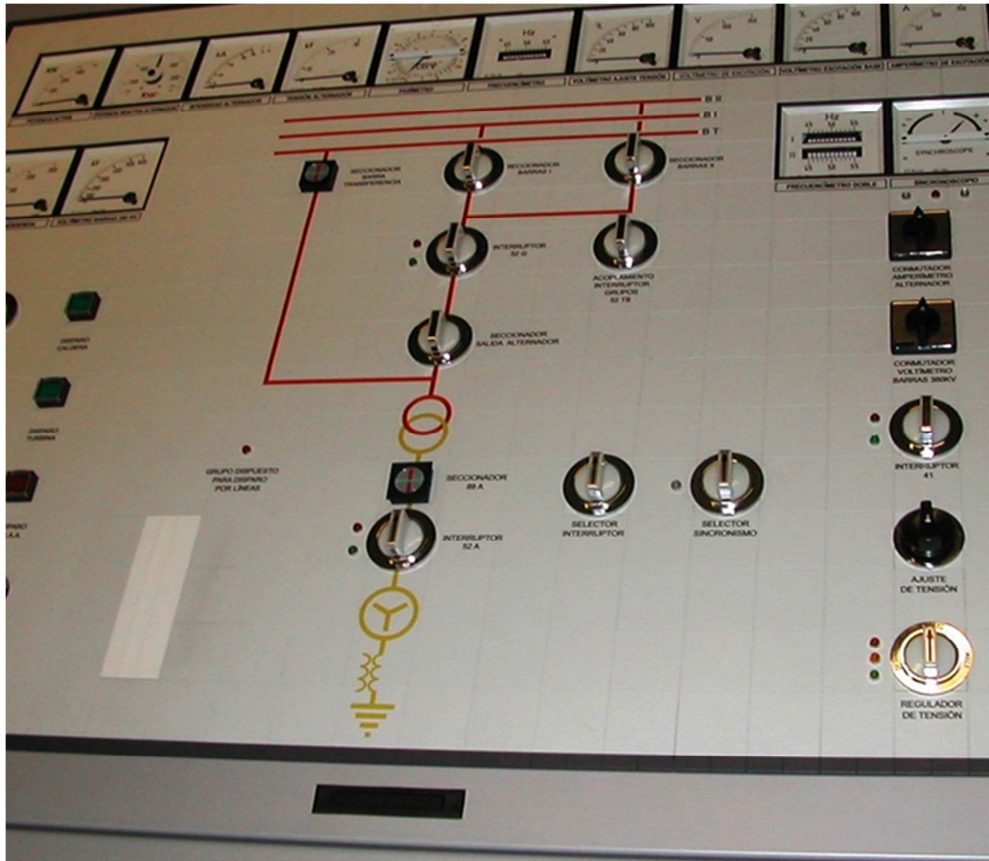


Figura 3: Panel sinóptico

Si nos ceñimos a la era moderna, las necesidades de ver en la distancia y controlar una máquina aparecen en los primeros cuadros de control, donde multitud de luces indicaban las diferentes situaciones previstas de la máquina. Cualquier situación imprevista, o pasada por alto, podía significar varias horas de trabajo de electricista para llevar la señal olvidada al panel de control y podía ser que no hubiera espacio para colocar el nuevo indicador.

La aparición de la informática permitió realizar este tipo de control de manera más sencilla. Ahora ya no sería necesario tener a verdaderos expertos en sistemas de automatización cada vez que hiciera falta cambiar el ajuste de un temporizador en un sistema de control.

Los grandes cuadros de control empezaban a convertirse en monitores que podían mostrar la misma información. Pero cualquier cambio en la presentación era más sencillo de realizar. Bastaban unas modificaciones en el código de la aplicación para que en la pantalla apareciera, por ejemplo, un contador de piezas olvidado en el momento de realizar el diseño del ejemplo anterior.

Vista la necesidad, varios fabricantes desarrollaron entonces paquetes de software capaces de comunicarse con los sistemas de control existentes y permitieron así una flexibilidad de uso no imaginada hasta el momento. Esta tendencia ha ido en aumento, de tal manera que hoy día las opciones existentes son numerosísimas. Algunos de los más conocidos:

- |               |       |
|---------------|-------|
| - Intellution | IFIX  |
| - Omron       | SCS   |
| - Siemens     | WinCC |

- Rockwell Automation                      RS-View
- Wonderware                                InTouch
- GE-Fanuc                                    Cimplicity

La evolución de los sistemas operativos ha incrementado también las posibilidades de estos sistemas, permitiendo las estructuras multipuesto gracias a los sistemas de red informáticos.

Con la irrupción de Internet en el mundo de las comunicaciones industriales ahora es posible conectarse con un sistema de control situado en cualquier lugar del mundo gracias a la tecnología Web-Server: un ordenador dotado de un explorador y la dirección IP del sistema que queremos visualizar serán suficientes.

### **2.3. Interfaz Hombre-Máquina (HMI).**

A mediados de los noventa aparece una versión de *Visual BASIC* que permite crear, con gran facilidad, controles gráficos e interfaces de usuario gracias a utilidades ya definidas. Colocar un botón en pantalla era simplemente eso, picar y arrastrar el botón, ya confeccionado, hasta el lugar deseado. A esto hay que añadirle el desarrollo una década antes de las técnicas de intercambio de información entre aplicaciones (*DDE* del inglés *Dynamic data Exchange*). De esta forma, las primeras interfaces hombre-máquina comenzaron a ser posibles, facilitando así la representación y supervisión de los sistemas de control.

Podemos definir estas interfaces como el aparato que presenta los datos a un operador humano y a través del cual éste controla el proceso. La industria del *HMI* nació esencialmente de la necesidad de estandarizar la manera de monitorear y controlar múltiples sistemas remotos, *PLC's* y otros mecanismos de control. Un HMI puede tener vínculos también con una base de datos para proporcionar las tendencias, los datos de diagnóstico y el manejo de la información así como un cronograma de procedimientos de mantenimiento, información logística, esquema detallados para un sensor o máquina, incluso sistemas expertos con guía de resolución de problemas. En la *Figura 6* podemos ver un ejemplo de estas interfaces:



*Figura 6: Interfaz Hombre-Máquina en un cuadro eléctrico*

Debemos tener en cuenta también a la hora de definir las *HMI*, que estas interfaces son incapaces de supervisar un proceso completo de cualquier sistema de control. Las *HMI* nos

ofrecen información del estado de dicho proceso o lo que es lo mismo, podremos monitorizar el estado de la actividad o proceso a la que se dedique la interfaz.

Otra limitación de los sistemas *HMI* se debe a que carecen de una base de datos con la que manejar mucha información. Esta limitación viene dada ya que por lo general las *HMI* se utilizan para configurar paneles con una memoria y capacidad de almacenamiento de datos limitada.

Normalmente las *HMI* hoy día se encuentran como ayuda a los sistemas de control generales en cualquier instalación o en cualquier proceso a controlar. Aunque también se pueden encontrar conectadas a un *PLC* a través de un *PC* y una tarjeta de comunicación desarrollando funciones de control secundarias.

Por último, cabe destacar que cada fabricante ofrece su propio paquete de software para configurar dichas interfaces. Tomando como ejemplo la compañía *Siemens AG*, la empresa germana tiene como programas principales para elaborar las *HMI* la herramienta *Protool* y el programa *WinCC flexible*. Como dato curioso, añadir que este último software también nos da la posibilidad de diseñar sistemas *SCADAS*, que ocuparan el siguiente punto de este segundo capítulo.

## 2.4. Los sistemas SCADA.

### 2.4.1. Introducción y contexto.

Podríamos decir que el principal objetivo de la automatización industrial consiste en gobernar la actividad y la evolución de los procesos sin la intervención continua de un operador humano.

En los últimos años, se ha estado desarrollado un sistema, denominado *SCADA*, el cual permite supervisar y controlar, las distintas variables que se encuentran en un proceso o planta determinada. Para ello se deben utilizar distintos periféricos, softwares de aplicación, unidades remotas, sistemas de comunicación, etc... los cuales permiten al operador mediante la visualización en una pantalla de computador, tener el completo acceso al proceso.

Existen como sabemos varios sistemas que permiten controlar y supervisar, como lo son: *PLC*, *DCS* y ahora *SCADA*, que se pueden integrar y comunicar entre sí, mediante una red *Ethernet*, y así mejorar en tiempo real, la interfaz al operador.

Ahora no sólo se puede supervisar el proceso, sino además tener acceso al historial de las alarmas y variables de control con mayor claridad, combinar bases de datos relacionadas, presentar en un simple computador, por ejemplo, una plantilla *Excel*, documento *Word*, todo en ambiente *Windows*, siendo así todo el sistema más amigable.

### 2.4.2. Descripción de un sistema SCADA.

La palabra *SCADA* proviene del acrónimo *Supervisory, Control And Data Acquisition* que en castellano significaría *Supervisión, Control y Adquisición de Datos*.

Definimos un sistema *SCADA* como una aplicación o conjunto de aplicaciones software especialmente diseñada para funcionar sobre ordenadores de control de producción, con acceso a la planta mediante la comunicación digital con los instrumentos y actuadores, e interfaz gráfica de alto nivel con el usuario (pantallas táctiles, ratones o cursores, lápices ópticos...). Aunque inicialmente solo era un programa que permitía la supervisión y adquisición de datos en procesos de control, en los últimos tiempos han ido surgiendo una

serie de productos hardware y buses especialmente diseñados o adaptados para éste tipo de sistemas. La interconexión de los sistemas *SCADA* también es propia, se realiza una interfaz del *PC* a la planta centralizada, cerrando el lazo sobre el ordenador principal de supervisión.

El sistema permite comunicarse con los dispositivos de campo (controladores autónomos, autómatas programables, sistemas de dosificación, etc.) para controlar el proceso en forma automática desde la pantalla del ordenador, que es configurada por el usuario y puede ser modificada con facilidad. Además, provee de toda la información que se genera en el proceso productivo a diversos usuarios.

Los sistemas *SCADA* se utilizan en el control de oleoductos, sistemas de transmisión de energía eléctrica, yacimientos de gas y petróleo, redes de distribución de gas natural, subterráneos, generación energética (convencional y nuclear)...

No todos los sistemas *SCADA* están limitados a procesos industriales sino que también se ha extendido su uso a instalaciones experimentales como la fusión nuclear o los colisionadores del *CERN* en Suiza por añadir más ejemplos.

### **2.4.3. Objetivos de un SCADA.**

Entre los objetivos de estos sistemas podemos destacar:

- **Economía:** es más fácil ver qué ocurre en la instalación desde la oficina que enviar a un operario a realizar la tarea. Ciertas revisiones se convertirán en innecesarias.
- **Accesibilidad:** un parque eólico al completo (velocidad de cada rotor, producción de electricidad), lo tenemos en un clic de ratón encima de la mesa de trabajo. Será posible modificar los parámetros de funcionamiento de cada aerogenerador, poniendo fuera de servicio los que den indicios de anomalías; consultar el estado de las estaciones transformadoras del parque, detener los molinos que no sean necesarios, etc.
- **Mantenimiento:** la adquisición de datos materializa la posibilidad de obtener datos de un proceso, almacenarlos y presentarlos de manera inteligible para un usuario no especializado. La misma aplicación se puede programar de manera que nos avise cuando se aproximen las fechas de revisión o cuando una máquina tenga más fallos de los considerados normales.
- **Ergonomía:** es la ciencia que procura hacer que la relación entre el usuario y el proceso sea lo menos tirante posible. Los modernos ordenadores con sus prestaciones gráficas, intentan sustituir a los grandes paneles, repletos de cables, pilotos y demás elementos informativos.
- **Gestión:** todos los datos recopilados pueden ser valorados de múltiples maneras mediante herramientas estadísticas, gráficas, valores tabulados, etc., que permitan explotar el sistema con el mejor rendimiento posible.
- **Flexibilidad:** cualquier modificación de alguna de las características del sistema de visualización (añadir el estado de un contador de piezas, realizar algún cálculo) no significa un gasto en tiempo y medios, pues no hay modificaciones físicas que requieran la instalación de un cableado o del contador.
- **Conectividad:** se buscan sistemas abiertos, es decir, sin secretos ni sorpresas para el integrador. La documentación de los protocolos de comunicación actuales permite la



interconexión de sistemas de diferentes proveedores y evita la existencia de lagunas informativas que puedan causar fallos en el funcionamiento o en la seguridad.

#### **2.4.4. Características de los sistemas SCADA.**

Sistemas de control hay muchos y muy variados y todos, bien aplicados, ofrecen soluciones óptimas en entornos industriales. Lo que hace de los sistemas *SCADA* una herramienta diferenciada es la característica de **control supervisado**. De hecho, la parte de control viene definida y supeditada, por el proceso a controlar, y en última instancia, por el hardware e instrumental de control (*PLC's*, controladores lógicos, armarios de control...) o los algoritmos lógicos de control aplicados sobre la planta los cuales pueden existir previamente a la implantación del sistema *SCADA*, el cual se instalará sobre y en función de estos sistemas de control.

En consecuencia, supervisamos el control de la planta y no solamente monitorizamos las variables que en un momento determinado están actuando sobre la planta; esto es, podemos actuar y variar las variables de control en tiempo real, algo que pocos sistemas permiten con la facilidad intuitiva que dan este tipo de sistemas.

De este modo se puede realizar una inspección superior de todo el conjunto o de algunas partes en particular y examinar algún detalle nuevamente para corregirlo, de forma que el sistema funcione según nuestras exigencias. El supervisor (el operario a cargo de dicha tarea) que efectúa una labor determinante, podrá garantizar el perfecto funcionamiento del proceso, aumentando así su rendimiento. En resumidas cuentas, con los sistemas *SCADA*, el supervisor u operario es capaz de variar el proceso a la misma vez que éste se está desarrollando. Es un control supervisado a tiempo real. Esto diferencia notablemente los sistemas *SCADA* de los sistemas clásicos de automatización donde las variables de control están distribuidas sobre los controladores electrónicos de la planta y dificulta mucho una variación en el proceso de control, ya que estos sistemas una vez implementados no permiten un control a tiempo real óptimo.

Como en los sistemas *HMI*, los sistemas *SCADA* también nos permiten **monitorear** todo el proceso en su conjunto así como otras acciones dentro del mismo proceso. Es decir, los sistemas de automatización de interfaz gráfica tipo *HMI* básicos, ofrecen una gestión de alarmas en formato rudimentarias mediante las cuales la única opción que le queda al operario es realizar una parada de emergencia, reparar o compensar la anomalía y realizar un *reset*. En los sistemas *SCADA*, se utiliza un *HMI* interactivo el cual permite detectar alarmas y a través de la pantalla solucionar el problema mediante las acciones adecuadas en tiempo real. Esto otorga una gran flexibilidad a los sistemas *SCADA*. En definitiva, el modo supervisor del *HMI* de un sistema *SCADA* no solamente señala los problemas, sino lo más importante, orienta en los procedimientos para solucionarlos. Ciertamente es por otro lado, que todos los sistemas *SCADA* ofrecen una interfaz gráfica *PC-Operario* tipo *HMI*, pero no todos los sistemas de automatización que tienen *HMI* son *SCADA*. La diferencia radica en la función de supervisión que pueden realizar estos últimos a través del *HMI*. A través de las interfaces hombre-máquina, los *SCADA* pueden desempeñar las siguientes funciones:

- Adquisición y almacenado de datos, para recoger, procesar y almacenar la información recibida, en forma continua y confiable.
- Representación gráfica y animada de variables de proceso y monitorización de éstas por medio de alarmas.
- Ejecutar acciones de control, para modificar la evolución del proceso, actuando bien sobre los reguladores autónomos básicos (consignas, alarmas, menús, etc.) bien directamente sobre el proceso mediante las salidas conectadas.

- Arquitectura abierta y flexible con capacidad de ampliación y adaptación.
- Conectividad con otras aplicaciones y bases de datos, locales o distribuidas en redes de comunicación.
- Supervisión, para observar desde un monitor la evolución de las variables de control.
- Transmisión, de información con dispositivos de campo y otros *PC*.
- Base de datos, gestión de datos con bajos tiempos de acceso. Suele utilizar *ODBC*.
- Presentación, representación gráfica de los datos. Interfaz del Operador o *HMI* (*Human Machine Interface*).
- Explotación de los datos adquiridos para gestión de la calidad, control estadístico, gestión de la producción y gestión administrativa y financiera.
- Alertar al operador de cambios detectados en la planta, tanto aquellos que no se consideren normales (alarmas) como cambios que se produzcan en la operación diaria de la planta (eventos). Estos cambios son almacenados en el sistema para su posterior análisis.

#### **2.4.5. Prestaciones.**

Estas son las prestaciones destacables que nos ofrecen los sistemas *SCADA*:

- Posibilidad de crear paneles de alarma, que exigen la presencia del ordenador para reconocer una parada o situación de alarma, con registro de incidencias.
- Generación de históricos de señal de planta, que pueden ser volcados para su proceso sobre una hoja de cálculo.
- Creación de informes, avisos y documentación en general.
- Ejecución de programas, que modifican la ley de control, o incluso el programa total sobre el autómata (bajo unas ciertas condiciones) .
- Posibilidad de programación numérica, que permite realizar cálculos aritméticos de elevada resolución sobre la *CPU* del ordenador, y no sobre la del autómata, menos especializado, etc.

Con todo ello se pueden desarrollar aplicaciones basadas en el *PC*, con captura de datos, análisis de señales, presentaciones en pantalla, envío de resultados a disco o impresora, control de actuadores, etc.

#### **2.4.6. Requisitos.**

Además de las prestaciones, debemos tener en cuenta también los requisitos necesarios que todo sistema *SCADA* debe cumplir para considerarse como tal:

- Deben ser sistemas de arquitecturas abiertas, capaces de crecer o adaptarse según las necesidades cambiantes de la empresa.
- Deben comunicarse con total facilidad y de forma transparente para el usuario con el equipo de planta (“drivers”) y con el resto de la empresa (acceso a redes locales y de gestión).
- Los programas deberán ser sencillos de instalar, sin excesivas exigencias, y fáciles de utilizar, con interfaces amables con el usuario (sonido, imágenes, pantallas táctiles...).

#### **2.4.7. Ventajas.**

Después de conocer las prestaciones, los requisitos y las características de estos sistemas, es hora de enumerar, las ventajas que un *SCADA* puede ofrecernos a la hora de controlar y supervisar cualquier proceso, del tipo que sea. Si bien algunas de sus ventajas ya se han explicado en puntos anteriores, ahora vamos a enumerarlas de forma concisa para mayor entendimiento.

Cuando hablamos de un sistema *SCADA* no hay que olvidar que hay algo más que las pantallas que nos informan de cómo van las cosas en nuestra instalación. Tras éstas se encuentran multitud de elementos de regulación y control, sistemas de comunicaciones y múltiples utilidades de software que pretenden que el sistema funcione de forma eficiente y segura.

Estas serían las ventajas más significativas que nos ofrecen este tipo de sistemas de control y supervisión:

- El actual nivel de desarrollo de los paquetes de visualización permite la creación de aplicaciones funcionales sin necesidad de ser un experto en la materia.
- Un sistema *PLC* está concebido para trabajar en condiciones adversas, proporcionando robustez y fiabilidad al sistema que controla.
- La modularidad de los autómatas permite adaptarlos a las necesidades actuales y ampliarlos posteriormente si es necesario.
- Cualquier tipo de sensores y actuadores puede integrarse en un programa de *PLC* mediante las múltiples tarjetas de adquisición disponibles (tensión, corriente, sondas de temperatura, etc.).
- Gracias a las herramientas de diagnóstico se consigue una localización más rápida de errores. Esto permite minimizar los periodos de paro en las instalaciones y repercute en la reducción de costes de mantenimiento.
- Un sistema de control remoto o unidad terminal remota (*RTU*) puede definirse de manera que pueda funcionar de forma autónoma, aún sin comunicaciones con la estación maestra.

- El concepto de tele-mantenimiento permite realizar modificaciones de software en las estaciones remotas (*RTU*) desde el centro de control.
- Los programas de control pueden documentarse convenientemente de manera que puedan ser fácilmente interpretados por los técnicos de mantenimiento.
- Un conjunto de manuales de usuario y documentación técnica adecuados permitirán el manejo satisfactorio por terceras personas.
- Los sistemas de diagnóstico implementados en los elementos de control informan continuamente de cualquier incidencia en los equipos.
- Los programas de visualización pueden presentar todo tipo de ayuda al usuario, desde la aparición de una alarma hasta la localización de la causa o la parte de esquema eléctrico implicada en la misma. Esto permite reducir los tiempos de localización de averías al proporcionarse información sobre el origen y las causas de los fallos.
- Generación y distribución automática de documentación. El sistema de visualización puede recoger los datos del autómatas y presentarlos en formatos fácilmente exportables a otras aplicaciones de uso común, tales como hojas de cálculo.
- Haciendo uso de las tecnologías celulares (*GSM*, *GPRS*, *UMTS*), los sistemas de control pueden mantener informados sobre cualquier incidencia a los operadores responsables de los mismos mediante mensajes de correo electrónico o de voz.
- La integración de sistemas es rápida gracias a los sistemas de comunicación estandarizados.
- La tecnología Web permite el acceso desde cualquier punto geográfico a nuestro sistema de control.
- Los protocolos de seguridad permiten una gestión segura y eficiente de los datos, limitando el acceso a personas no autorizadas.
- Aumento de calidad del producto mediante las herramientas de diagnóstico. El operador es notificado en el momento en que se detecta una incidencia.
- La reducción de personal permite menor número de equipos de mantenimiento, más reducidos y mejor coordinados gracias a la información proveniente de las estaciones remotas, evaluada en el centro de control.
- Posibilidad de mantenimiento por parte de suministradores locales de servicios.
- El nivel de descentralización va en aumento, apostando por la modularidad. Esto permite una mayor disponibilidad, pues las funciones de control se pueden repartir y/o duplicar.
- La distribución de recursos y control sobre la red permite una mejor coordinación entre las estaciones remotas en caso de fallos en una de ellas.
- Mediante las redes de comunicación, el sistema *SCADA* se integra en la red corporativa, permite la integración entre los niveles de *Campo* y *Gestión* y completa así la estructura *CIM* (*Computer Integrated Manufacturing*).

### 2.4.8. Estructura hardware de los sistemas SCADA.

Aunque los sistemas SCADA se configuren mediante un programa informático, como todo software, también necesita apoyarse en un sistema físico o estructura hardware. En este punto vamos a representar y explicar la estructura hardware básica necesaria para un sistema SCADA.

Podemos dividir el sistema hardware en tres partes bien diferenciadas, y una cuarta parte que hace referencia a la red o redes de comunicación necesarias para interconectar todo el hardware. Estas partes son:

- **Nivel de gerencia.**
- **Nivel de automatización.**
- **Instrumentos de campo.**
- **Red de comunicación.**

En la *Figura 7*, a continuación, podemos observar cada una de las partes que componen el entramado hardware de un SCADA.

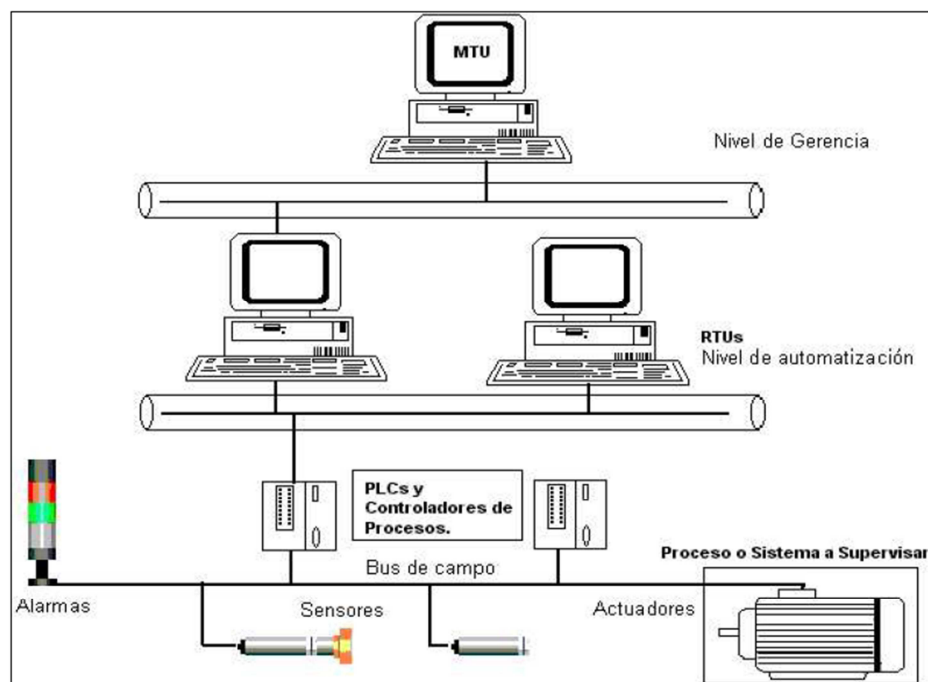


Figura 7: Estructura básica de un sistema SCADA

Esta integración por niveles es similar a la pirámide de la automatización, conocida como CIM, de sus siglas en inglés *Computer Integrated Manufacturing*. A continuación vamos a describir cada uno de esos niveles.

**Nivel de gerencia:** en este nivel encontramos al ordenador central o MTU (*Master Terminal Unit*). Se trata del ordenador principal del sistema el cual supervisa y recoge la información del resto de las subestaciones, bien sean otros ordenadores conectados (en sistemas complejos) a los instrumentos de campo o directamente sobre dichos instrumentos. Este

ordenador suele ser un *PC*, el cual soporta el *HMI*. De esto se deriva que el sistema *SCADA* más sencillo es el compuesto por un único ordenador, el cual es el *MTU* que supervisa toda la estación. Las funciones principales de la *MTU* son:

- Interroga en forma periódica a las *RTU's*, y les transmite consignas; siguiendo usualmente un esquema maestro-esclavo.
- Actúa como interfaz al operador, incluyendo la presentación de información de variables en tiempo real, la administración de alarmas, y la recolección y presentación de información en forma de historial de sucesos.
- Puede ejecutar software especializado que cumple funciones específicas asociadas al proceso supervisado por el *SCADA*. Por ejemplo, software para detección de pérdidas en un oleoducto.

**Nivel de automatización:** en este nivel se hallan los ordenadores remotos o *RTU*. Estos ordenadores están situados en los nodos estratégicos del sistema gestionando y controlando las subestaciones del sistema, reciben las señales de los sensores de campo, y comandan los elementos finales de control ejecutando el software de la aplicación *SCADA*. Se encuentran en el nivel intermedio o de automatización, a un nivel superior está el *MTU* y a un nivel inferior los distintos instrumentos de campo que son los que ejercen la automatización física del sistema, control y adquisición de datos.

Estos ordenadores no tienen por qué ser *PC's*, ya que la necesidad de soportar un *HMI* no es tan grande a este nivel, por lo tanto suelen ser ordenadores industriales tipo armarios de control, aunque en sistemas muy complejos pueden haber subestaciones intermedias en formato *HMI*.

Una tendencia actual es la de dotar a los *PLC's* (en función de las *E/S* a gestionar) con la capacidad de funcionar como *RTU's* gracias a un nivel de integración mayor y *CPU's* con mayor potencia de cálculo. Esta solución minimiza costes en sistemas donde las subestaciones no sean muy complejas sustituyendo el ordenador industrial mucho más costoso.

En la *Figura 8* podemos observar la estructura básica de un *RTU* en relación a las entradas/salidas (*E/S*) del sistema.

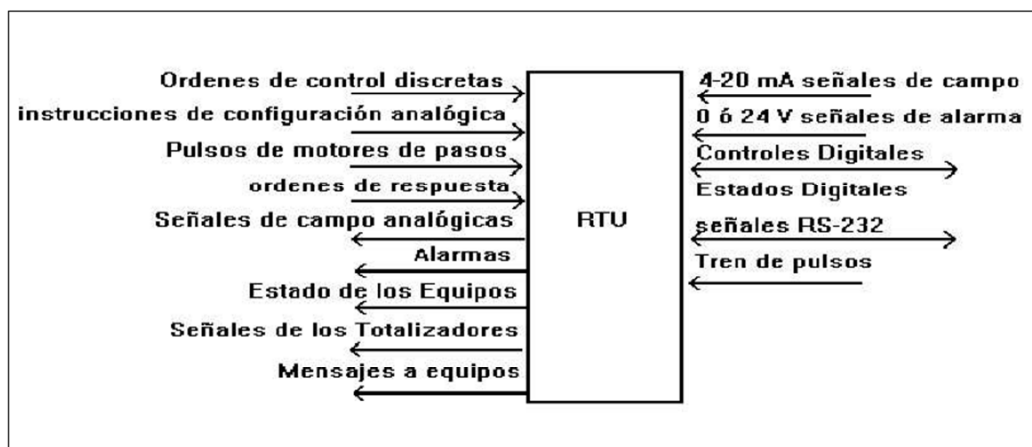


Figura 8: Estructura básica de un *RTU* en base a las *E/S* del sistema

**Instrumentos de campo:** son todos aquellos que permiten tanto realizar la automatización o control del sistema (*PLC's*, controladores de procesos industriales, y actuadores en general) como los que se encargan de la captación de información del sistema (sensores y alarmas).

**Red de comunicación:** este es el nivel que gestiona la información que los instrumentos de campo envían a la red de ordenadores desde el sistema. El tipo de bus utilizado en las comunicaciones puede ser muy variado según las necesidades del sistema y del software escogido para implementar el sistema *SCADA*, ya que no todos los softwares (así como los instrumentos de campo como *PLC's*) pueden trabajar con todos los tipos de bus.

Hoy en día, gracias a la estandarización de las comunicaciones con los dispositivos de campo, podemos implementar un sistema *SCADA* sobre prácticamente cualquier tipo de bus. Podemos encontrar *SCADA's* sobre formatos estándares como los *RS-232*, *RS-422* y *RS-485* a partir de los cuales, y mediante un protocolo *TCP/IP*, podemos conectar el sistema sobre un bus en configuración *DMS* ya existente; pasando por todo tipo de buses de campo industriales, hasta formas más modernas de comunicación como *Bluetooth* (bus de radio), micro-ondas, satélite, cable...

Aparte del tipo de bus, existen interfaces de comunicación especiales para la comunicación en un sistema *SCADA* como puede ser módems para estos sistemas que soportan los protocolos de comunicación *SCADA* y facilitan la implementación de la aplicación.

Otra característica de las comunicaciones de un sistema *SCADA* es que la mayoría se implementan sobre sistemas *WAN* de comunicaciones, es decir, los distintos terminales *RTU* pueden estar separados geográficamente.

#### **2.4.9. ¿Cuándo implementar un sistema SCADA?**

Una vez que ya tenemos claro en qué consiste un sistema *SCADA*, es hora de conocer cómo y cuándo se reúnen las condiciones necesarias para implementar estos sistemas. Como es obvio, va a depender en mayor parte, del proceso que queramos supervisar. Por lo tanto, la pauta a seguir la marcará dicho proceso.

En primer lugar, debemos tener en cuenta el número de variables a representar gráficamente en el sistema de control. Si el proceso que queremos controlar consta de un número elevado de variables, es un candidato idóneo para poner en marcha un sistema *SCADA*.

Otro aspecto importante es la distribución geográfica del sistema a controlar. Si nuestro sistema consta de un proceso o de varios procesos de fabricación distribuidos en distintas áreas geográficas, el *SCADA* es una herramienta muy útil para supervisar cada uno de los procesos.

Y por último y más importante, la información del proceso es necesario obtenerla en tiempo real. Como ya hemos apuntado en este capítulo, una de las grandes ventajas de estos sistemas de supervisión es la capacidad de obtener la información necesaria en tiempo real para poder realizar los ajustes requeridos en ese mismo intervalo.

Estas son las condiciones básicas para plantearnos la utilización de un sistema *SCADA* de control. A esto hay que añadirle las múltiples ventajas y características que se han citado en puntos anteriores.

#### 2.4.10. Implementando un SCADA.

Cuando una empresa decide implementar un sistema SCADA sobre su instalación hay 5 fases básicas a tener en cuenta para llevar a cabo el proceso:

- **Fase 1:** El diseño de la arquitectura del sistema. Esto incluye todas las consideraciones importantes sobre el sistema de comunicaciones de la empresa (tipo de bus de campo, distancias, número de *E/S*, protocolo del sistema y drivers...). También se verán involucrados los tipos de dispositivos que no están presentes en la planta pero que serán necesarios para supervisar los parámetros deseados.
- **Fase 2:** Equipamiento de la empresa con los *RTU's* necesarios, comunicaciones, equipos *HMI* y hardware en general. Adquisición de un paquete software *SCADA* adecuado a la arquitectura y sistemas de la planta.
- **Fase 3:** La instalación del equipo de comunicación y el sistema *PC*.
- **Fase 4:** Programación, tanto del equipamiento de comunicaciones como de los equipos *HMI* y software *SCADA*.
- **Fase 5:** Testeo del sistema o puesta a punto, durante el cual los problemas de programación en comunicaciones como en el software *SCADA* son solucionados.

#### 2.4.11. El software SCADA.

Un programa del tipo *HMI* se ejecuta en un ordenador o terminal gráfico y unos programas específicos le permiten comunicarse con los dispositivos de control de planta (hacia abajo) y los elementos de gestión (hacia arriba). Estos programas son lo que denominamos controladores (o *driver*) de comunicaciones tal y como vemos en la *Figura 9*:

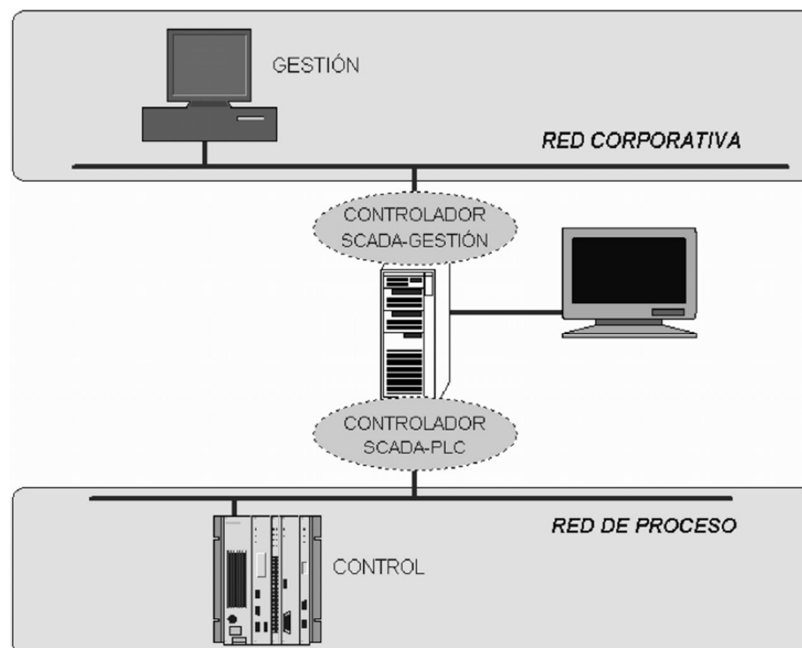


Figura 9: Concepto de driver o controlador



Una parte del paquete (propia o de terceros) contiene todos los controladores de comunicación entre nuestra aplicación y el exterior, ocupándose de gestionar los enlaces de comunicación, tratamiento de la información a transferir y protocolos de comunicación (*Profibus, AS-i, Can, Ethernet...*).

El *driver* realiza la función de traducción entre el lenguaje del programa *SCADA* y el del autómatas (hacia abajo, por ejemplo, *Profibus*), o entre el *SCADA* y la red de gestión de la empresa (hacia arriba, con *Ethernet*, por ejemplo).

Generalmente la configuración del controlador de comunicaciones se realiza durante la instalación del software principal o como programa de acceso externo al ejecutar la aplicación principal.

Según la importancia del sistema, es posible especializar componentes, realizando tareas exclusivas dentro del sistema de control (servidores de datos, de alarmas, de históricos, de interfaz hombre-máquina, etc.).

Una vez los datos de planta se han procesado, pueden transferirse a otras aplicaciones de software, tales como hojas de cálculo o bases de datos. Esto es lo que podríamos denominar gestión de datos, que nos permite analizar eventos, alarmas, emergencias, etc., ocurridos durante la producción.

En un programa *SCADA* tendremos dos bloques bien diferenciados: el programa de desarrollo y el programa de ejecución o *Run-Time*.

- El programa de desarrollo engloba las utilidades relacionadas con la creación y edición de las diferentes ventanas de la aplicación, así como sus características (textos, dibujos, colores, propiedades de los objetos, programas, etc.).
- El programa *Run-Time* permite ejecutar la aplicación creada con el programa de desarrollo (en Industria se entrega, como producto acabado, el *Run-Time* y la aplicación).

En definitiva, las características del software para configurar un sistema *SCADA* deben ser las siguientes:

- Manejo del soporte o canal de comunicación.
- Manejo de uno o varios protocolos de comunicación (drive).
- Manejo y actualización de una base de datos.
- Administración de alarmas (eventos).
- Generación de archivos históricos.
- Interfaces con el operador (*MMI -Man Machine Interface-* y *HMI -Human Machine Interface-*).
- Capacidad de programación (*Visual Basic, C*).
- Transferencia dinámica de datos (*DDE*).
- Conexión a redes.

- Debe tener capacidad para comunicarse con múltiples redes de instrumentos, aun siendo de distinta procedencia y fabricantes (estándar IEC 1131.3).

En la *Figura 10* puede verse de forma más detallada todo lo expuesto anteriormente donde se ven bien diferenciados los terminales remotos (*RTU's*) y de gestión central (*MTU's*).

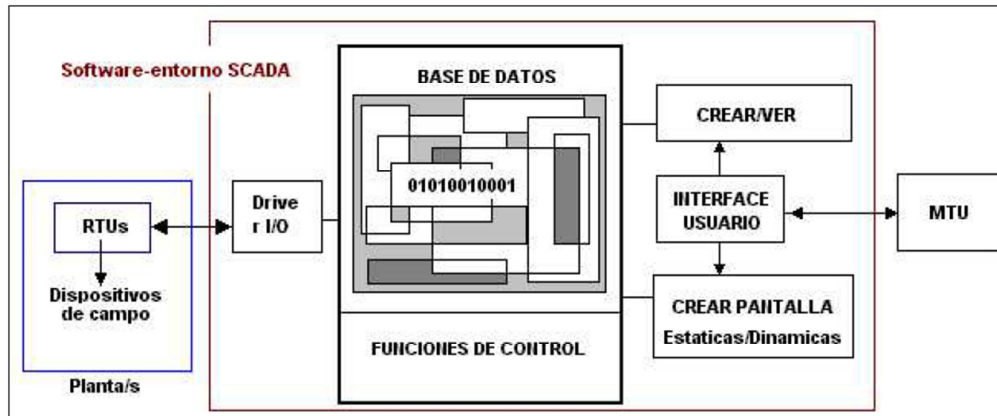


Figura 10: Entorno de un software SCADA

#### 2.4.12. Estructura del software SCADA.

Podríamos dividir la estructura del software SCADA en cuatro módulos que realizan una función concreta. Son los siguientes:

- **Configuración.**
- **Interfaz gráfica del operador.**
- **Módulo de proceso.**
- **Gestión y archivo de datos.**

**Configuración:** permite al usuario definir el entorno de trabajo de su aplicación según la disposición de pantallas requerida y los niveles de acceso para los distintos usuarios.

Dentro del módulo de configuración el usuario define las pantallas gráficas o de texto que va a utilizar, importándolas desde otra aplicación o generándolas desde el propio SCADA. Para ello, se incorpora un editor gráfico que permite dibujar a nivel de píxel (punto de pantalla) o utilizar elementos estándar disponibles, líneas, círculos, textos o figuras, con funciones de edición típicas como copiar, mover, borrar, etc. tal y como muestra la *Figura 11*.

También durante la configuración se seleccionan los *drivers* de comunicación que permitirán el enlace con los elementos de campo y la conexión o no en red de estos últimos, se selecciona el puerto de comunicación sobre el ordenador y los parámetros de la misma, etc.

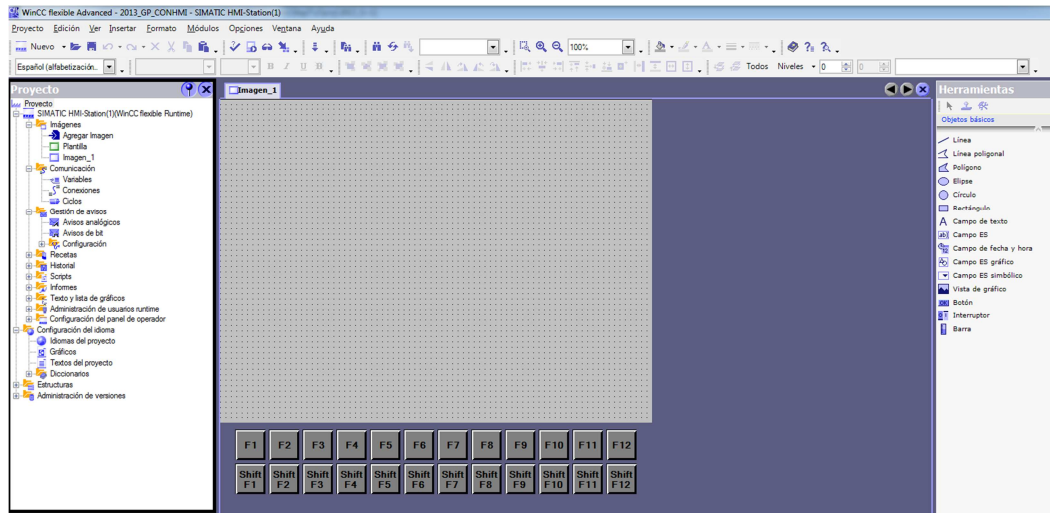


Figura 11: Interfaz de configuración de WinCC flexible

**Interfaz gráfica del operador:** proporciona al operador las funciones de control y supervisión de la planta.

El proceso a supervisar se representa mediante sinópticos gráficos almacenados en el ordenador de proceso y generados desde el editor incorporado en el SCADA o importados desde otra aplicación de uso general (*Paintbrush*, *DrawPerfect*, *AutoCAD*, etc.) durante la configuración del paquete.

Los sinópticos están formados por un fondo fijo y varias zonas activas que cambian dinámicamente a diferentes formas y colores, según los valores leídos en la planta o en respuesta a las acciones del operador.

Se tienen que tener en cuenta algunas consideraciones a la hora de diseñar las pantallas:

- Las pantallas deben tener apariencia consistente, con zonas diferenciadas para mostrar la planta (sinópticos), los botones y entradas de mando (control) y las salidas de mensajes del sistema (estados, alarmas).
- La representación del proceso se realizará preferentemente mediante sinópticos que se desarrollan de izquierda a derecha.
- La información presentada aparecerá sobre el elemento gráfico que la genera o soporta, y las señales de control estarán agrupadas por funciones.
- La clasificación por colores ayuda a la comprensión rápida de la información. Los colores serán usados de forma consistente en toda la aplicación: si *rojo* significa peligro o alarma, y *verde* se percibe como indicación de normalidad, éste será el significado dado a estos colores en cualquier parte de la aplicación.

La *Figura 12* muestra una interfaz gráfica completa de un sistema SCADA donde se tienen en cuenta las especificaciones anteriormente expuestas:

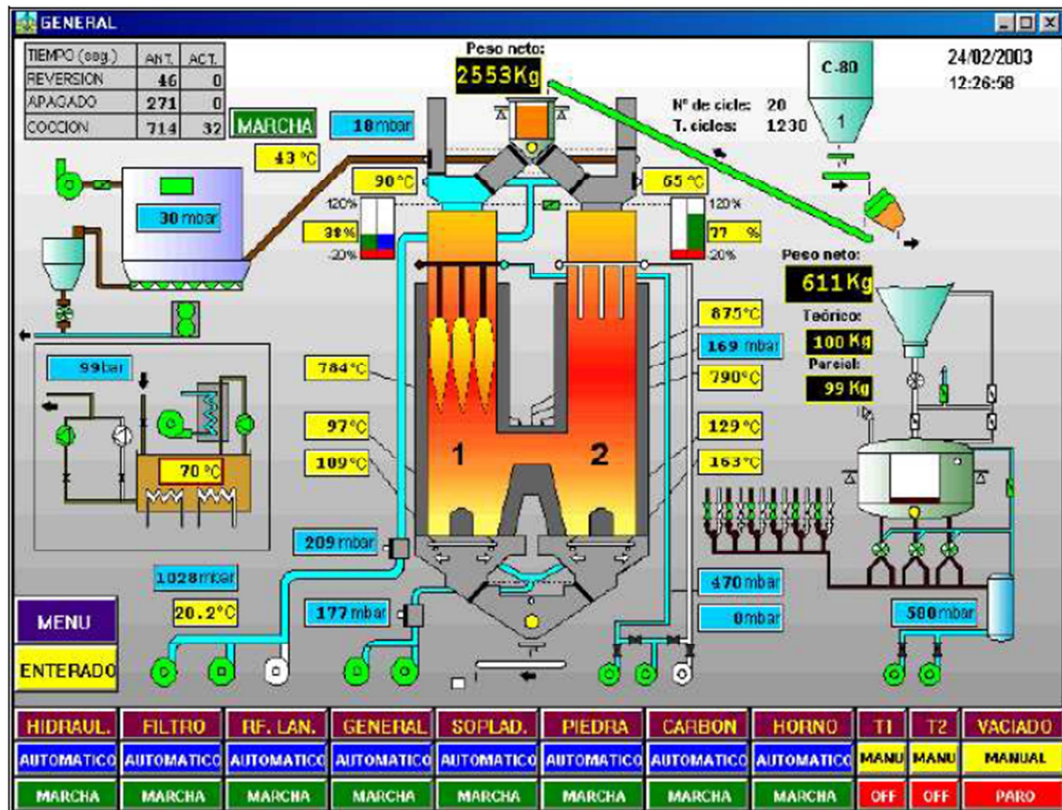


Figura 12: Interfaz gráfica del operador

**Módulo de proceso:** ejecuta las acciones de mando pre-programadas a partir de los valores actuales de variables leídas.

Sobre cada pantalla se puede programar relaciones entre variables del ordenador o del autómatas que se ejecutan continuamente mientras la pantalla esté activa. La programación se realiza por medio de bloques de programa en lenguaje de alto nivel (C, Basic, etc.).

Es muy frecuente que el sistema SCADA confíe a los dispositivos de campo, principalmente autómatas, el trabajo de control directo de la planta, reservándose para sí las operaciones propias de la supervisión, como el control del proceso, análisis de tendencias, generación de históricos, etc.

Las relaciones entre variables que constituyen el programa de mando que el SCADA ejecuta de forma automática pueden ser de los tipos siguientes:

- Acciones de mando automáticas pre-programadas dependiendo de valores de señales de entrada, salida o combinaciones de éstas.
- Maniobras o secuencias de acciones de mando.
- Animación de figuras y dibujos, asociando su forma, color, tamaño, etc., al valor actual de las variables.
- Gestión de recetas, que modifican los parámetros de producción (consignas de tiempo, de conteo, estados de variables, etc.) de forma pre-programada en el tiempo o dinámicamente según la evolución de planta.

**Gestión y archivo de datos:** se encarga del almacenamiento y procesado ordenado de los datos, según formatos inteligibles para periféricos hardware (impresoras, registradores) o software (bases de datos, hojas de cálculo) del sistema, de forma que otra aplicación o dispositivo pueda tener acceso a ellos.

Pueden seleccionarse datos de planta para ser capturados a intervalos periódicos, como un registro histórico de actividad, o para ser procesados inmediatamente por alguna aplicación software para presentaciones estadísticas, análisis de calidad o mantenimiento. Esto último se consigue con un intercambio de datos dinámico entre el SCADA y el resto de aplicaciones que corren bajo el mismo sistema operativo.

Por ejemplo, el protocolo *DDE* de *Windows* permite intercambio de datos en tiempo real. Para ello, el SCADA actúa como un servidor *DDE* que carga variables de planta y las deja en memoria para su uso por otras aplicaciones *Windows*, o las lee en memoria para su propio uso después de haber sido escritas por otras aplicaciones.

Una vez procesados, los datos se presentan en forma de gráficas analógicas, histogramas, representación tridimensional, etc., que permiten después analizar la evolución global del proceso.

La siguiente imagen (*Figura 13*) muestra las gráficas de rendimiento de un proceso controlado por un SCADA.

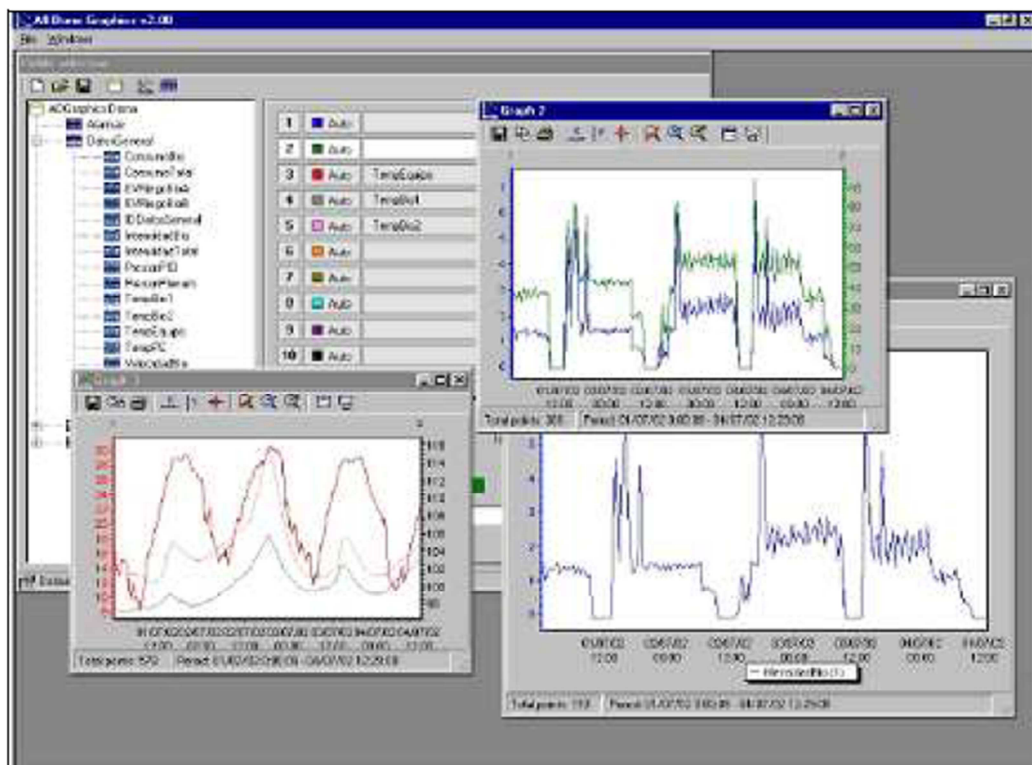


Figura 13: Gráficas con un SCADA

### 2.4.13. Comunicaciones.

Las interfaces de comunicación permiten al ordenador de gestión o *MTU* acceder a los dispositivos de campo, a través de los *RTU's*. Así, la interfaz de comunicación enlazará el *MTU* con los distintos *RTU's* del sistema a través del bus de campo como muestra la *Figura 14*.

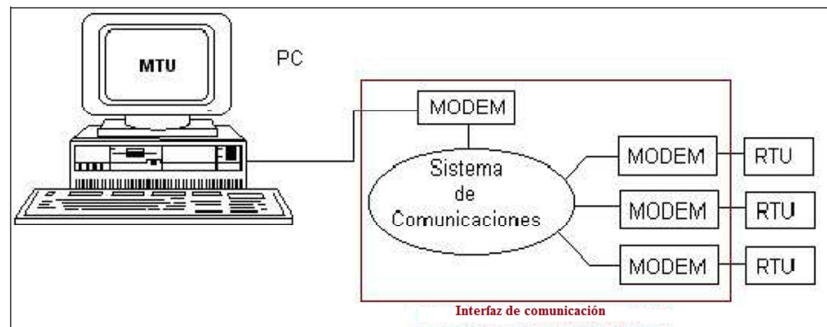


Figura 14: Interfaz de comunicación

Las interfaces de comunicación constan de distintos elementos que permiten el intercambio de información entre el *MTU* y los *RTU's*:

- La base del sistema de comunicación es el bus de campo que es el que transporta la información y las órdenes de control; éste vendrá definido en función del tamaño del sistema *SCADA* (número de *E/S* del sistema), distancias entre *RTU's* y/o disponibilidad del servicio público de comunicación (para sistemas *SCADA* de tipo red *WAN* en interconexión entre distintas plantas).
- Los módem que conectan físicamente los *RTU's* y el *MTU* al bus.
- El módulo de comunicaciones contiene los drivers de conexión con el resto de elementos digitales conectados, entendiendo el driver como un programa (software) que se encarga de la iniciación del enlace, aplicación de los formatos, ordenación de las transferencias, etc., en definitiva, de la gestión del protocolo de comunicación. Estos protocolos pueden ser abiertos (*ModBus*, *FieldBus*, *Map*, etc.), o propios de fabricante.

Estos drivers, propios del software *SCADA*, deben comunicarse con otros paquetes de software por medio de *DDE* (*Dynamic Data Exchange*) y *DLL* (*Dynamic Link Libraries*) como canal de comunicación, implementados por el sistema operativo, que permite que diversos paquetes de software envíen y reciban datos comunes. Por ejemplo se puede relacionar una celda de una hoja de cálculo con una variable del sistema y así variar puntos de consignas del proceso, o bien comunicación directa con los drivers de *I/O* de los dispositivos de campo. En la imagen siguiente (*Figura 15*) se muestra la arquitectura de los drivers de un *SCADA*.



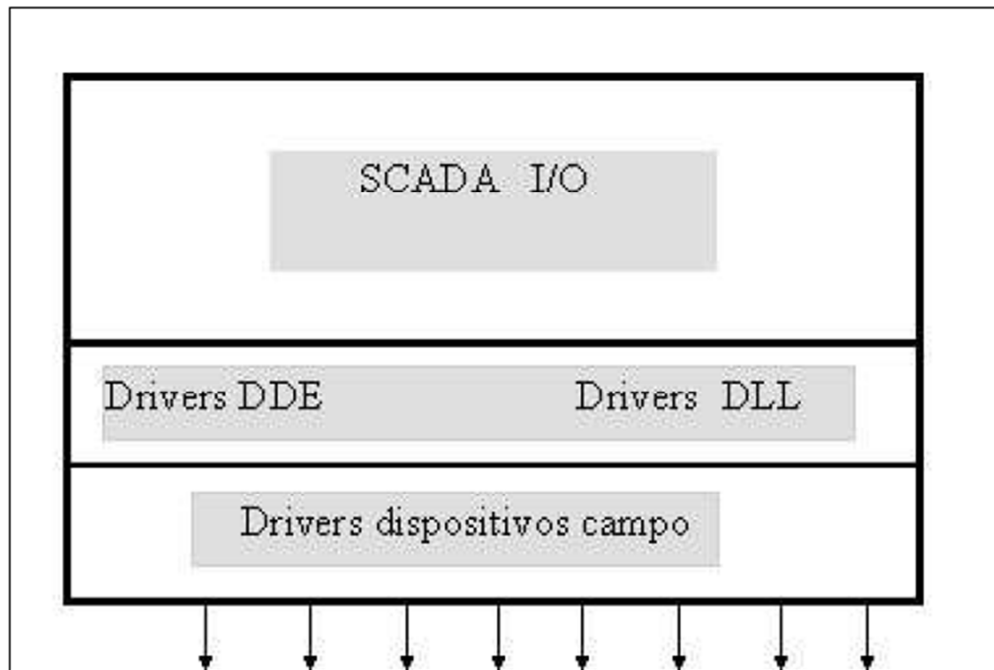


Figura 15: Diagrama de la arquitectura de los drivers de un SCADA

Adicionalmente, y en los SCADA distribuidos en arquitecturas cliente-servidor, los módulos de comunicaciones son también los responsables del enlace entre los diferentes ordenadores de proceso que soportan la aplicación, con un enlace probablemente establecido sobre una red local *DECnet*, *TCP/IP*, *IPX/SOX*, *NETBIOS*, *MAP/TOP*, *Novell*, etc.

#### 2.4.14. Comunicaciones entre aplicaciones.

Los métodos de intercambio de información entre aplicaciones informáticas más conocidos son:

**OPC:** El estándar de intercambio de datos por excelencia se denomina *OPC (OLE for Process Control)*. Es un estándar abierto que permite un método fiable para acceder a los datos desde aparatos de campo. El método de acceso siempre es el mismo, sin depender del tipo y origen de los datos.

Se basa en la tecnología *COM (Component Object Model)*, de *Microsoft*, que permite definir cualquier elemento de campo mediante sus propiedades, convirtiéndolo en una interfaz. De esta manera es posible conectar fácilmente cualquier elemento de campo con un servidor de datos local (*COM*), o remoto (*DCOM*).

Los componentes *OPC* se pueden clasificar en clientes o servidores:

- Cliente *OPC (OPC client)*: es una aplicación que sólo utiliza datos, tal como hace un paquete SCADA. Cualquier cliente *OPC* se puede comunicar con cualquier servidor *OPC* sin importar el tipo de elemento que recoge esos datos (el aspecto que veremos, desde el punto de vista de los datos, será siempre similar, sin importar el fabricante del equipo).
- Servidor *OPC (OPC server)*: es una aplicación que realiza la recopilación de datos de los diversos elementos de campo de un sistema automatizado y permite el acceso libre a estos elementos desde otras aplicaciones que los soliciten (clientes *OPC*).

Los productos *OPC* (clientes y servidores), pueden ser usados con *Visual Basic* y sus variantes. Es decir, *OPC* corresponde a un conjunto de especificaciones basadas en los estándares de Microsoft (*COM*, *DCOM*, *OLE Automation*, y *ActiveX*) que cubren los requerimientos de comunicación industrial entre aplicaciones y dispositivos, especialmente en lo que se refiere a la atención al tiempo real. En la *Figura 16* observamos la interfaz *OPC* servidor-cliente.

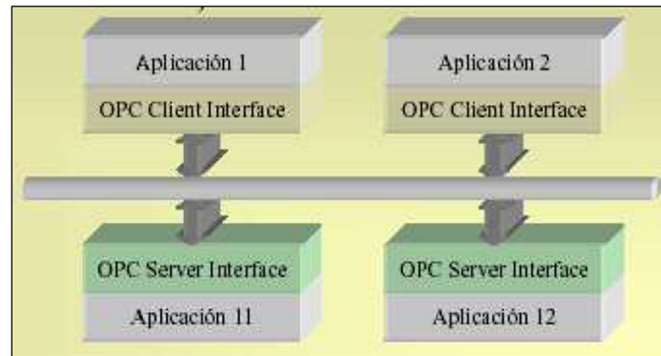


Figura 16: Interfaz OPC servidor-cliente

Mediante *OLE Automation* podemos comunicar las aplicaciones con datos recibidos a través de *LAN*, estaciones remotas o Internet.

Antes del *OPC*, cada software requería de una interfaz específico (servidor, driver) para intercambiar datos con una determinada familia de dispositivos de campo. Y para intercambiar datos entre aplicaciones se utilizaba el estándar *DDE* o bien interfaces específicos para cada pareja de aplicaciones. *OPC* elimina esta necesidad y permite utilizar una misma operativa para comunicar aplicaciones, dispositivos y drivers. Podemos observar en la *Figura 17* dos diagramas que muestran la arquitectura *OPC*.

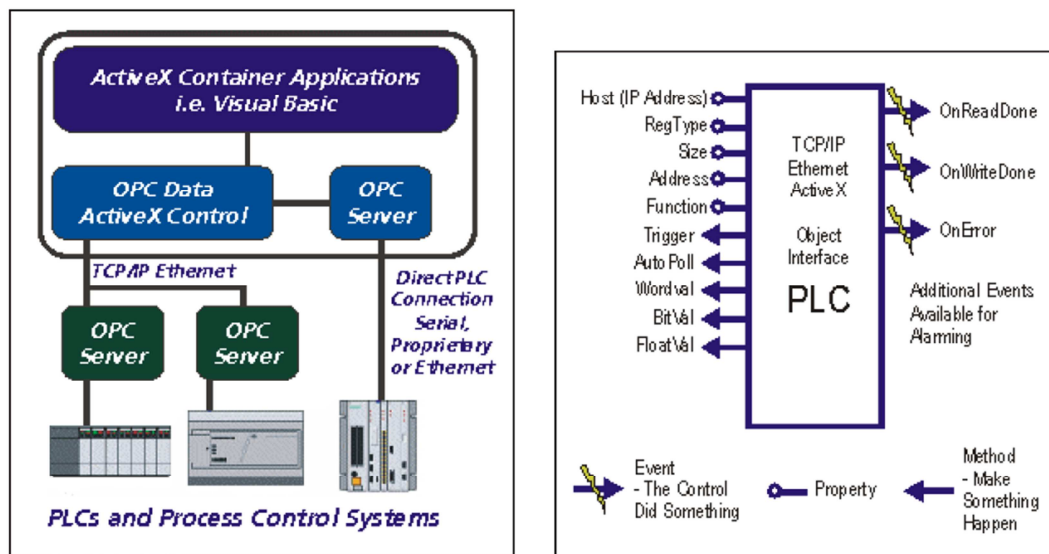


Figura 17: Arquitectura OPC



**Conectividad remota WebServer (a través de *Internet*):** algunas de las ventajas de la utilización de *Internet* en los entornos SCADA son el ofrecimiento de una funcionalidad total, ofreciendo su operatividad a través de cualquier navegador estándar. La información en tiempo real de la planta de proceso es inmediatamente accesible para cualquier persona autorizada de la organización, esté donde esté, con el coste más bajo. En la siguiente imagen (Figura 18) podemos ver el esquema de una conexión a través de *Internet*.

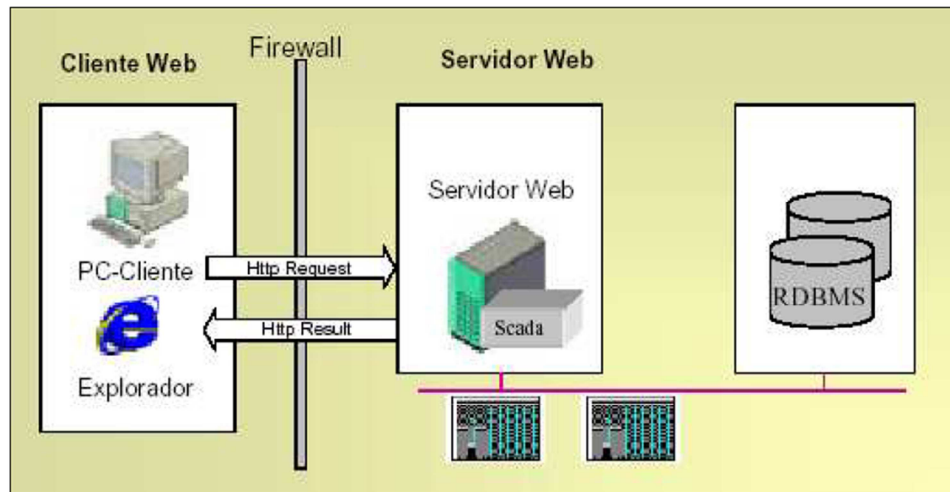


Figura 18: Conexión a través de Internet

**ODBC:** mediante *ODBC (Open Data Base Connectivity)*, también de *Microsoft Windows*, tenemos un estándar que permite a las aplicaciones el acceso a datos que se encuentran en *Sistemas de Gestión de Bases de Datos (Data Base Management Systems)* utilizando *SQL* como método estándar de acceso.

*ODBC* permite que una aplicación pueda acceder a varias bases de datos mediante la inclusión del controlador correspondiente en la aplicación que debe acceder a los datos.

La interfaz *ODBC* define:

- Una librería de llamadas a funciones *ODBC*.
- La sintaxis *SQL* necesaria.
- Códigos de error estándar.
- El método de conexión a un *Sistema de Gestión de Bases de Datos (DBMS)*.
- El formato de presentación de los datos.

Para acceder a los datos, una aplicación necesita un controlador, que en *Windows* se llama *Librería de Enlace Dinámico (DLL, Dynamic Link Library)*. *ODBC* permite definir un estándar que permita el intercambio entre bases de datos y aplicaciones.

**SQL (*Structured Query Language*):** permite una interfaz común para el acceso a los datos por parte de cualquier programa que se ciña al estándar *SQL*. Las posibilidades de esta tecnología incluyen:

- Procedimientos: son bibliotecas de comandos almacenados en la base de datos. Permiten reducir el tráfico de red y simplificar los procedimientos de acceso a los usuarios de las bases de datos.
- Eventos: son comandos que se activan de forma automática bajo unas ciertas condiciones, facilitando el mantenimiento de la integridad de los datos.
- Replicación: permite la duplicación y sincronización de bases de datos. Por ejemplo, para actualizar los datos de la base de datos central con los almacenados en una unidad remota (*RTU*), más actuales, o para actualizar un servidor de datos que ha quedado temporalmente fuera de servicio y se vuelve a poner en funcionamiento.
- Accesibilidad: permite el intercambio o envío de información basándose en eventos. Por ejemplo, el envío automático de mensajes cuando se cumplen ciertas condiciones dentro de un sistema.

**ASCII:** mediante el formato *ASCII*, común a prácticamente todas las aplicaciones informáticas, tenemos un estándar básico de intercambio de datos. Es sencillo exportar e importar datos de configuración, valores de variables, etc.

**API:** las herramientas *API (Application Programming Interfaces)* permiten que el usuario pueda adaptar el sistema a sus necesidades mediante rutinas de programa propias escritas en lenguajes estandarizados, tales como *Visual Basic*, *C++*, o *Java*, lo cual les confiere una potencia muy elevada y una gran versatilidad. Permiten el acceso a las bases de datos de los servidores (valores almacenados temporalmente o archivos históricos).

#### **2.4.15. Evolución del software.**

En los últimos años ha existido una evolución de los productos software de supervisión y control para *PC (SCADA)* orientada a ampliar su campo de aplicación. De una supervisión y control iniciales a nivel de máquina o de proceso se ha pasado a una supervisión y control a nivel de planta. De una adquisición y registro de datos orientada a un control de proceso o de línea se ha ampliado su utilidad a proveer información en tiempo real del estado de la planta o de la fábrica.

El software orientado inicialmente a supervisión y control de proceso (máquina, proceso y línea) fue aprovechado para ampliar su utilidad a la supervisión y control de la producción. La adopción de forma generalizada de los estándares *COM/DCOM*, *Activ X*, *OPC* y *ODBC*, entre otros, por parte de la gran mayoría de proveedores facilitaba que los datos adquiridos mediante la aplicación *SCADA* estuvieran disponibles para otras aplicaciones como gestión de almacenes, *ERP*, etc.

Una de las demandas más generalizadas y, al mismo tiempo, una de las más críticas, es la capacidad de efectuar consultas trabajando con datos procedentes de diferentes fuentes: de diferentes aplicaciones (*SCADA*, *ERP*, etc.) o de bases de datos distintas y ubicadas en diferentes puntos del sistema.

Disponer del conjunto de drivers necesario para intercomunicar los diversos componentes de la solución completa, configurarlos y activarlos de forma transparente, es un elemento esencial para disponer de una integración efectiva.

Actualmente, diversos proveedores ofrecen módulos específicos orientados al almacenamiento de grandes cantidades de datos, así como servidores de datos capaces de atender consultas de grandes cantidades de datos y que implican tanto a datos recogidos de

proceso como a datos almacenados en otras bases de datos, y aptos para servir a múltiples usuarios, conectados a una red para la gestión y el control de la fábrica.

Por su naturaleza, estos módulos pueden llegar a constituir aplicaciones aptas para trabajar con *SCADAS* de otros fabricantes, servidores de datos históricos y servidores de datos integrados (procedentes de diferentes bases de datos o aplicaciones pero interrelacionados). Entre estos productos podemos citar: *IndustrialSQL Server* de *Wonderware*, *RSSql* de *Rockwell Software*, *Historian* de *GE Fanuc-Intellution*, etc.

Una de las necesidades del resultado de esta ampliación del entorno de aplicación ha sido la necesidad de disponer de herramientas cómodas, simples y potentes para la generación de interfaces de usuario que les permita acceder a la información que es de su interés. Dado que el entorno físico donde se ubican estos usuarios también se amplía, el medio generalizado de comunicación es Internet y la aplicación más común es cualquiera de los navegadores más difundidos.

Por otra parte, es conveniente disponer de herramientas que ofrezcan a cualquier usuario la posibilidad de diseñar y configurar una web específica que les permita dialogar con el sistema de información y obtener los datos necesarios. Un ejemplo de este tipo de aplicación es el *infoAgent* de *GE Fanuc-Intellution*, un software de edición, configuración y activación de portales de Internet que proporciona un servidor y admite múltiples clientes.

### **Tendencias:**

La madurez de los productos software para la adquisición y registro de datos en tiempo real y la supervisión y control de procesos ofrecen una evolución en los siguientes ámbitos:

- Su integración en entornos completos para la gestión del negocio disponiendo de información de planta en tiempo real, control y tratamiento de datos, y supervisión y gestión global de la empresa. La existencia de aplicaciones *MES*, los servidores de datos y los servidores de web son una prueba de ello.
- En el tratamiento de los datos adquiridos en planta por parte de sistemas expertos que ofrecen funcionalidades de detección y diagnóstico de fallos. Son evidentes las ventajas que supone disponer de un sistema experto que, a partir de los datos adquiridos de planta tanto en proceso continuo como discontinuo, pueda aplicar un conjunto de reglas que ayude al personal de operación en planta a detectar los fallos o situaciones delicadas y a tener una diagnosis de las causas que lo provocan, así como conocer cuál es la correcta actuación a seguir.
- La mejora de las interfaces con el usuario con el empleo de entornos gráficos de alta calidad, la incorporación de elementos multimedia de audio y vídeo, la mejora de los sistemas operativos para incrementar las velocidades de respuesta, el empleo de software orientado a objeto, con diálogos conversacionales con programador y usuario, etc., todo ello soportado por un hardware cada vez más compacto, fiable, potente, de mayor ancho de bus y más rápido.

### **2.4.16. Comparativas entre software SCADA.**

No sólo hay un único software para configurar un sistema *SCADA*. Existen múltiples opciones que se adaptan a las necesidades de cada proceso y de cada control. A continuación vamos a comparar, mediante una tabla (*Figura 19*), los distintos programas informáticos para la realización de los *SCADA*.

En ella podemos observar el lenguaje de programación más utilizado frente a los propios de cada sistema es el *Visual Basic*, ya que permite una mayor integración mediante la tecnología ofertada por *Microsoft* y así poder aprovechar al máximo sus posibilidades. De esta manera, queda justificable que todos los sistemas a continuación expuestos utilizan la tecnología *OPC cliente-servidor* para la adquisición de datos.

A modo de diferenciación entre unos sistemas y otros, las características que comportan más relevancia son los drivers utilizados para la conexión con el *PLC*, el número de variables del proceso, la estructura sobre la que recorre el sistema, la comunicación con el *PLC*...

Software	CX-Supervisor	All-Data	InTouch	Vijeo Look	Win CC
Subministrador/Fabricante	Omron/Omron (UK)	Freixas i Ros, S.L./Freixas i Ros, S.L.(E)	Logitech, S.A./Wonderware (USA)	Schneider Electric/Schneider Electric (F)	Siemens/Siemens
Requisitos recomendados del sistema	Windows 2000/NT 4.0 (Service Pack 5) IBM PC o compatible (200Mhz pentium o superior) Espacio libre de disco duro 100MB 128MB RAM Tarjeta SVGA 1024x768	Windows 2000/XP Profesional Intel Pentium III/IV, AMD Athlon Disco duro 20GB, 128MB RAM Tarjeta SVGA 1024x768 de resolución	Windows 2000/XP Profesional Intel Pentium III o superior Espacio libre de disco duro 2GB 512MB RAM	Windows 2000/XP Profesional procesador: Celeron 500Mhz...1.8Ghz RAM: 256MB...1024MB Espacio libre de disco duro 4GB SVGA 1024x768 de resolución	Windows 95/98 i Windows NT4.0/2000 de microsoft CPU INTEL Pentium II 400 MHz RAM 128MB Resolución 1024 x 768 Espacio disponible: > 500 MB
Drivers para PLC's	OMRON: todo los PLC's	OMRON: Sysmac serie C MOELLER: PS4-200 SIEMENS: simatic S5,S7-200/300/400 TELEMECANIQUE: unitelway otros ...	ALLEN-BRADLEY SIEMENS MODICON OPTO 22 SQUARE D OMRON otros...	TELEMECANIQUE AEG MODICON MODICON SQUARE D	SIMATIC ALLEN-BRADLEY MITSUBISHI FETELMECANIQUE UNI-TELWAY GE-FANUC MODICON OMRON serie C otros...
Lenguajes de programación	Visual Basic/Java	Visual Basic	propio (basado en C)	VBA (Visual Basic for Application)	Visual Basic C ANSI-C
Precio (euros)	600	1000-4000	/	/	/
Control de usuarios	si	si	si	si	si
nº variables	ilimitado	ilimitado	64-64k	128, 512 ó 1024	128, 256, 1024 ó 64k
comunicación entre módulos PC	/	UDP/Ethernet	Ethernet	/	TCP/IP NetBEUI
estructura	un único PC o varios PC's en multitarea	varios PC's en multitarea	un único PC o varios PC's en multitarea	un único PC	varios PC's en multitarea
gestión de alarmas	si	si	si	si	si
comunicación con PLC	RS-232C RS-422 Ethernet	UDP sobre Ethernet	DEVICE-Net PPI/MPI Profibus RS-232, RS-485	Uni-Telway o Modbus sobre RS-232 o RS-485; TCP/IP sobre Ethernet	Profibus TCP/IP Bus-SIPART Canal-DLL Modbus
adquisición de datos	cliente OPC	OPC Servidor/Cliente	OPC Servidor/Cliente	OPC Factory Server (OFS)	OPC Servidor/Cliente
bases de datos	MS-Access SQL SERVER ODBC paraORACLE	MDB	Servidor SQL MDSE	Access 2000 MSDE	Sybase SQL
intercambio de datos en el PC	automatización OLE controles ACTIVE-X comunicaciones DDE ADO (Active Data Object)	controles ACTIVE-X DBE	controles ACTIVE_X comunicaciones DDE	controles ACTIVE_X	controles ACTIVE_X aplicaciones OLE I CUSTOM

Figura 19: Tabla comparativa de software SCADA

Con toda esta información ya podemos hacernos una idea de los sistemas *SCADA* y de su funcionamiento y utilidad a la hora de supervisar y controlar cualquier proceso industrial, del tipo que sea. Pero no sólo existe la “*filosofía SCADA*” a la hora de supervisar, controlar y regular cualquier sistema. De la misma forma que a la hora de monitorizar cualquier proceso, podíamos utilizar una interfaz hombre-máquina (*HMI*) explicada en el punto 2.3 de este capítulo, para realizar el control podemos aplicar otro sistema de control. En el siguiente punto vamos a comparar los sistemas de control *SCADA* y *DCS*.

## 2.5. Otras filosofías de control. *SCADA* y *PLC* frente a *DCS*.

Como ya hemos apuntado anteriormente, no existe una única filosofía de control. A los sistemas *SCADA* se le unen también los sistemas *DCS*. Vamos a comprender el funcionamiento de los sistemas *DCS* mediante las diferencias que los separan de los *SCADA*. Las siglas *DCS* son el acrónimo de *Distributed Control System* y su traducción al castellano Sistema de Control Distribuido.

Una de las diferencias más notables se encuentra en su arquitectura hardware. Si recordamos la arquitectura hardware básica de los *SCADA* tal y como muestra la *Figura 20*:

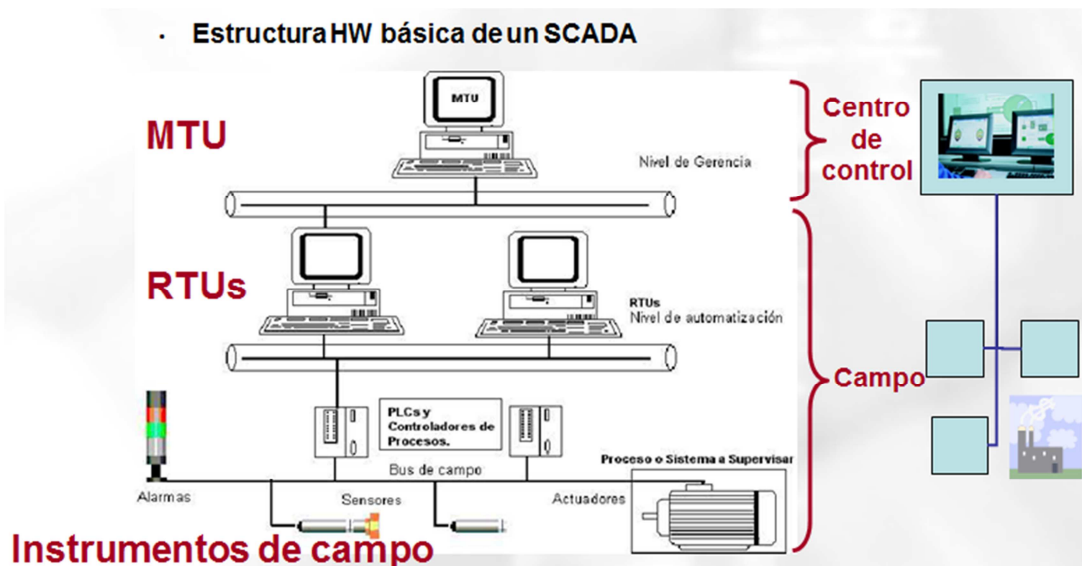


Figura 20: Arquitectura hardware de un SCADA

Podíamos encontrar tres partes bien diferenciadas. Una primera parte de control formada por la unidad *MTU*, la segunda parte de administración formada por los *RTU* y la tercera parte formada por los instrumentos de campo: sensores, *PLC*'s, etc... Dependiendo del proceso, el sistema de control podía constar simplemente de la primera y tercera parte formando así un sistema denominado *SCADA* y *PLC* tal y como vemos en la *Figura 21*:

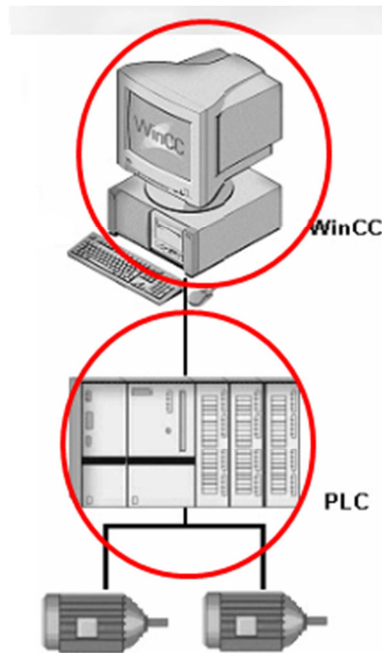


Figura 21: SCADA y PLC

En esta estructura, el *PC* se encarga de visualizar y controlar el proceso mediante alguno de los software *SCADA* diseñados para ordenador que se han visto anteriormente, mientras que el *PLC* o autómatas controla el proceso y actúa sobre el mismo dependiendo de la programación de su *CPU* y del operario que visualiza todo el proceso.

Por otro lado, la arquitectura hardware de los sistemas *DCS* es bien distinta a lo que hemos visto. Si observamos con detenimiento la *Figura 22* podremos ver las diferencias.

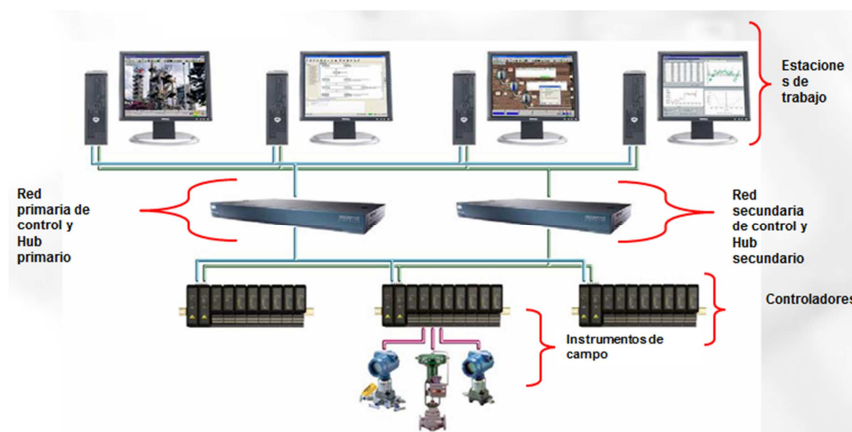


Figura 22: Arquitectura hardware de un DCS

En los *DCS*, debido a su arquitectura todos los equipos del sistema (ordenadores, servidores, controladores) están sincronizados contra un mismo reloj patrón, de forma que todas las medidas, alarmas y eventos tienen una misma marca de tiempo.

Además, todos los equipos están conectados a una gran base de datos a través de los *hub* primarios y secundarios, compartiendo la misma información. De esta forma, no sólo existe una

única unidad de control general, sino que da la posibilidad a disponer de más unidades principales de control.

En cuanto al número de variables que son capaces de implementar, los *DCS* superan por bastante a los *SCADA*.

Otra característica diferenciable de los *SCADA* y los *DCS* radica en la programación del sistema. Mientras que en los *SCADA* y *PLC* debemos programar por separado las interfaces de control y el autómatas, en los *DCS* la programación se hace mediante bloques de programación que generan a su vez el código necesario para los controladores y pantallas del operador. Además, las interfaces de usuario ya están predefinidas al estilo arrastra y suelta o *drag and drop*.

Por otro lado, el ciclo de duración de los *DCS* es mayor que el de los *SCADA*. Teniendo en cuenta este factor, el coste de implementación, mucho mayor en los *DCS*, se compensa con una vida útil más prolongada que en los sistemas *SCADA*.

La última gran diferencia está en el uso. Nos referimos a los sistemas y procesos que deben controlar. Debido a que los *DCS* son más caros por sus características particulares, éstos se emplean normalmente en sistemas e instalaciones industriales de gran tamaño, las cuales constan de un número grande de procesos y variables a configurar y representar. Quedando de esta forma los sistemas formados por un *SCADA* y un *PLC* para aplicaciones relativamente más pequeñas.

## **2.6. El SCADA de Siemens.**

En este capítulo hemos hecho un recorrido por los sistemas de control, enfocándonos en los *SCADA*, conociendo estos últimos y comprendiéndolos. Viendo sus características y ventajas, cuándo y cómo implementarlos e incluso comparándolos con otros sistemas de control. Pero al realizar este *Proyecto Final de Carrera* con dispositivos *Siemens*, este capítulo no estaría completo si no dedicáramos un apartado para explicar, de forma que se tenga una noción general, el sistema *SCADA* que nos ofrece la marca teutona.

El concepto principal sobre el que se ha basado *Siemens* a la hora de ofrecer una solución para los sistemas de visualización y control, es la integración. Con la integración, la marca alemana pretende reducir el software necesario al máximo.

De esta idea nace *TIA Portal*. *Totally Integrated Automation Portal* es el software que optimiza todos sus procedimientos de procesamiento, operación de máquinas y planificación. Con su intuitiva interfaz de usuario, la sencillez de sus funciones y la completa transparencia de datos es increíblemente fácil de utilizar. Los datos y proyectos preexistentes pueden integrarse sin ningún esfuerzo, lo cual asegura su inversión a largo plazo.

El *SCADA* de *Siemens* se basa en la arquitectura hardware de *SCADA* y *PLC* (*Figura 23*). Por lo tanto, los programas que se integran en la herramienta *TIA Portal* tienen una función bien definida. De esta manera se puede dividir el software en:

- Step 7: para programar el controlador o *PLC*
- WinCC: para programar las interfaces del sistema *SCADA*.



Figura 23: SCADA y PLC

Gracias a la integración, se consigue combinar estos dos programas en uno solo, agilizando y facilitando así el diseño e implementación de los sistemas SCADA. Su mayor ventaja está en la información que comparten, de forma que cualquier variable que configuremos en *Step 7* podrá ser asociada en *WinCC* sin necesidad de crear una nueva. Además también seremos capaces de integrar y configurar redes de comunicación, de forma que las direcciones que asociemos, variantes o características, serán compartidas por todo los aparatos de control del sistema SCADA.

Además, *Siemens* ofrece la posibilidad de integrar estos dos programas sin necesidad de usar la herramienta *TIA Portal*. Para ello, simplemente hay que instalar en el PC que servirá de control de gestión o MTU el software de programación de PLC, *Step 7* y a continuación el software de SCADA, *WinCC*<sup>1</sup>. De esta forma, ambos programas se integran de manera automática.

Con esta breve explicación del sistema SCADA de *Siemens*, concluye este capítulo basado en los sistemas de visualización y control.

<sup>1</sup> Ver anexos para la integración del software.



# Capítulo 3: La maqueta.

Dedicaremos este capítulo para explicar la configuración de la maqueta de pruebas de la que disponemos, así como de las modificaciones pertinentes en la misma para completar el sistema físico de nuestro proyecto. Además, como ya se ha comentado, el sistema físico consta de tres partes diferenciadas que son:

- La maqueta. Compuesta tres depósitos y sensorizados cada uno de ellos, se le ha acoplado el sensor inteligente de *Siemens*, *SITRANS Probe LU*. Para comunicar este dispositivo con el resto de componentes de control del sistema físico, se utiliza una pasarela *PROFIBUS DP/PA*.
- El autómatas o Controlador Lógico Programable que se encargará de activar o desactivar distintos componentes de la maqueta.
- La estación de trabajo o puesto de control desde donde se visualizará y controlará todo el proceso. Está compuesto por un *PC* con una tarjeta de comunicación y los correspondientes programas informáticos.

En la *Figura 1* podemos ver representado el sistema físico y su organización comunicativa mediante bloques después de realizar las modificaciones necesarias.

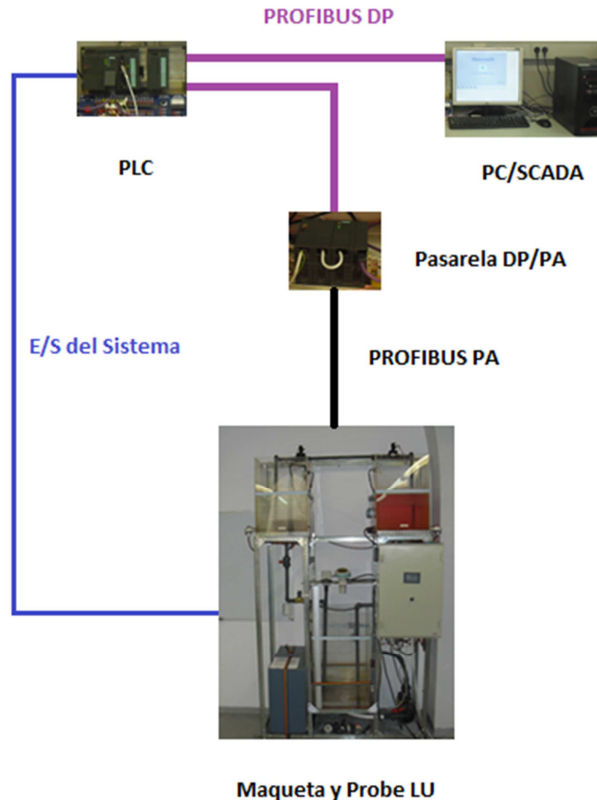
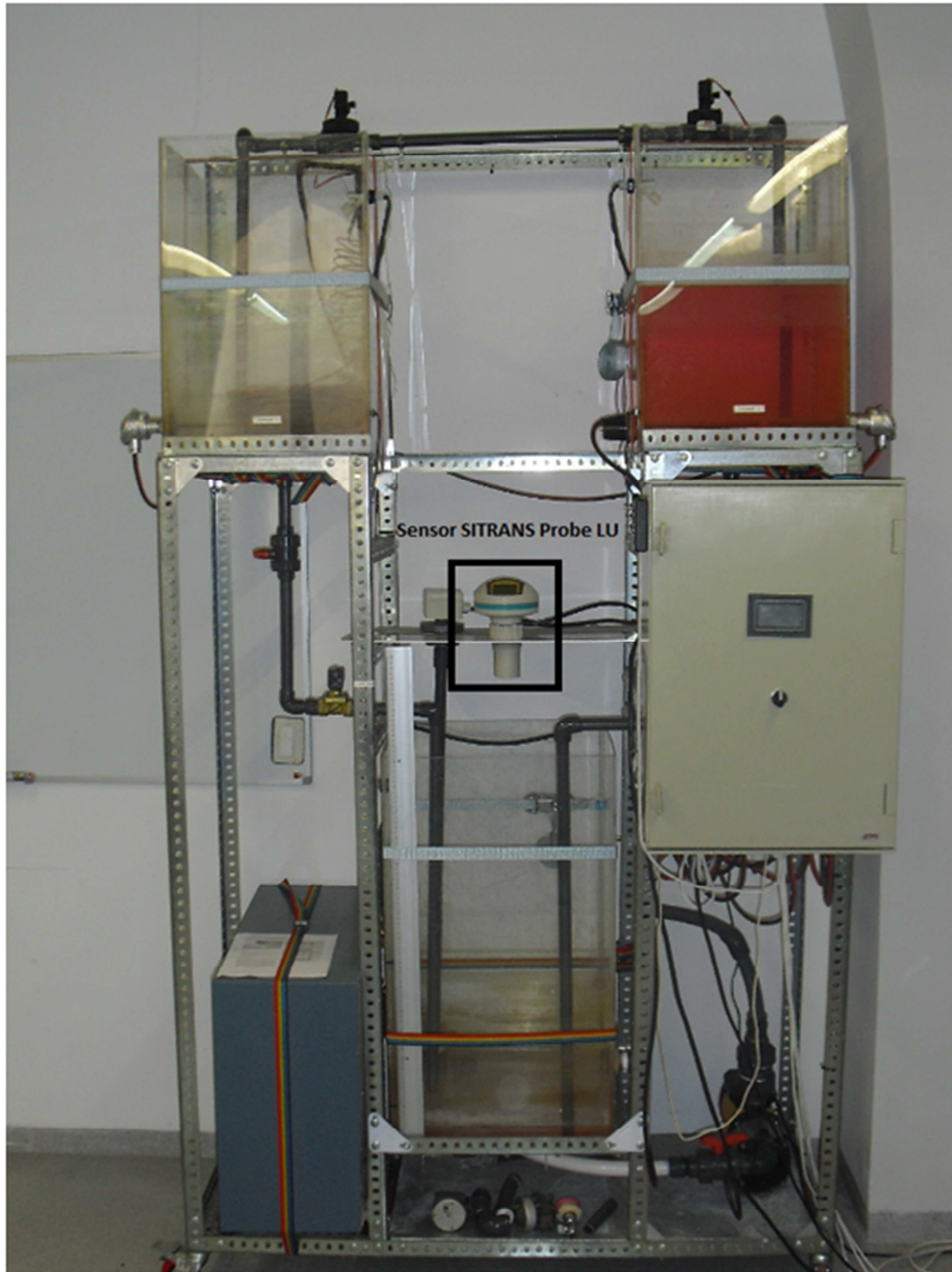


Figura 1: Sistema físico

### 3.1. La Maqueta.

La maqueta utilizada en nuestro sistema físico proviene de un anterior proyecto, el cual carecía de un sistema de control y supervisión (SCADA) y a la que se le ha acoplado para este trabajo, el sensor inteligente *SITRANS Probe LU* de Siemens, tal y como muestra la *Figura 2*.



*Figura 2: Maqueta con sensor SITRANS Probe LU*

### 3.1.1. Estructura metálica.

La estructura metálica conforma el armazón o esqueleto de la maqueta y soporta el peso del resto de elementos de la misma.

Consiste en una base metálica de unos  $0.75 \text{ m}^2$  ( $1.50 \times 0.5 \text{ m}$  aproximadamente) a la que se le han instalado cuatro ruedas, una por cada esquina, para facilitar su transporte. En ella se asientan la máquina de frío y la bomba del sistema. A partir de esta base se elevan los perfiles metálicos necesarios para fijar las peanas que sustentan los tres depósitos a controlar por el sistema.

Las peanas superiores tienen una base de  $0.25 \text{ m}^2$  ( $0.50 \times 0.50 \text{ m}$ ) y se elevan a una altura de  $1.60 \text{ m}$  sobre la base metálica. Están unidas mediante otro perfil transversal en su parte posterior. La peana inferior se eleva unos  $0.4 \text{ m}$  sobre la base metálica y ocupa un área de  $0.225 \text{ m}^2$  ( $0.45 \times 0.50$ ). Las uniones de los perfiles con las peanas han sido reforzadas, tal y como muestra la *Figura 3*.



*Figura 3: Unión reforzada*

Además sobre las peanas superiores se elevan sendos perfiles que unidos a otro transversal, soportan la instalación hidráulica superior con sus electroválvulas a una altura de  $2.25 \text{ m}$ . (*Figura 4*).



*Figura 4: Detalle del perfil superior transversal*

Por último, la estructura también soporta el peso de un armario que aloja el cuadro eléctrico situado debajo de la peana superior derecha como se observa en la *Figura 5*. El armario se sitúa a unos  $0.80 \text{ m}$ . de la base metálica.



*Figura 5: Armario eléctrico*

### **3.1.2. Instalación hidráulica.**

Además de su estructura metálica, otra parte básica de la maqueta es su instalación hidráulica. Se han utilizado tuberías de *PVC* de  $\varnothing 25$  mm ( $\frac{3}{4}$  de pulgada) y de  $\varnothing 32$  mm (1 pulgada) para las conexiones con la bomba a modo de optimizar su rendimiento. En la *Figura 6* podemos ver la instalación hidráulica del sistema:

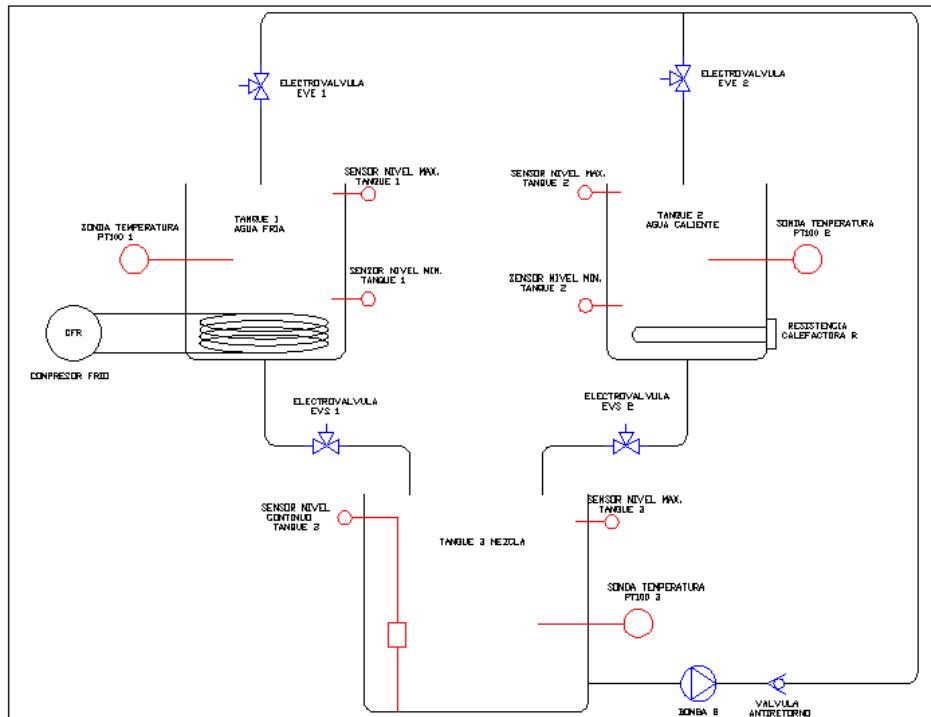


Figura 6: Esquema hidráulico

Como podemos observar en la imagen anterior, la instalación hidráulica está formada además por tres depósitos, una bomba y cuatro electroválvulas.

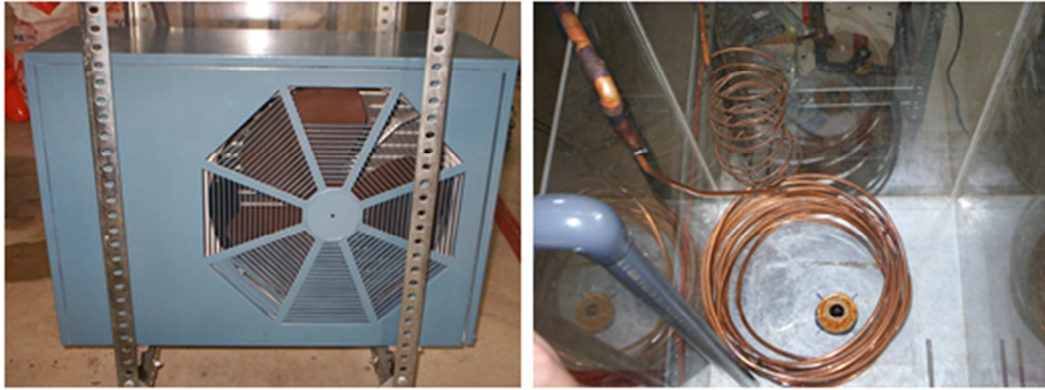
### Depósitos.

El líquido es contenido en tres depósitos de metacrilato, además de circular por cada uno de ellos. Cada depósito tiene una función asignada, siendo su cometido el siguiente:

- Depósito superior izquierdo: dedicado a enfriar su contenido. Tanque de frío con sensores<sup>1</sup> de nivel máximo y mínimo y de temperatura.
- Depósito superior derecho: dedicado a calentar su contenido. Tanque de calor con sensores de nivel máximo y mínimo y de temperatura.
- Depósito inferior central: contiene la mezcla de líquido proveniente de los depósitos superiores. Tanque de mezcla con sonda de nivel, sensor inteligente y de nivel máximo y de temperatura.

<sup>1</sup> Los sensores de la maqueta serán explicados más adelante en este capítulo.

El depósito superior izquierdo utiliza una máquina de aire acondicionado a la que se le ha instalado un serpentín de fabricación casera para enfriar su contenido. El serpentín consta de 12 m. de tubería de cobre de  $\varnothing$  127 mm ( $\frac{1}{2}$  pulgada) y 2 m. de tubería de  $\varnothing$  15.875 mm ( $\frac{5}{8}$  de pulgada) para construir los capilares de enfriamiento, como muestra la *Figura 7*:

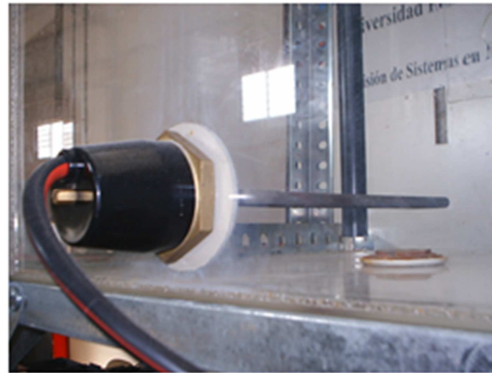


*Figura 7: a) Máquina de aire acondicionado*

*b) Serpentín*

La máquina de aire acondicionado comprime el gas y este fluye por el serpentín hasta llegar a la unión con los capilares. En la reducción debida a la unión entre las dos partes, el gas disminuye su presión. Al entrar al capilar, el gas se expande a baja presión, enfriándose y enfriando a su vez el capilar y este, el contenido del depósito.

Por otro lado, el depósito superior derecho se encarga de calentar su contenido. Para ello se ha colocado una resistencia calefactora de 750 vatios de potencia, como se observa en la *Figura 8*:

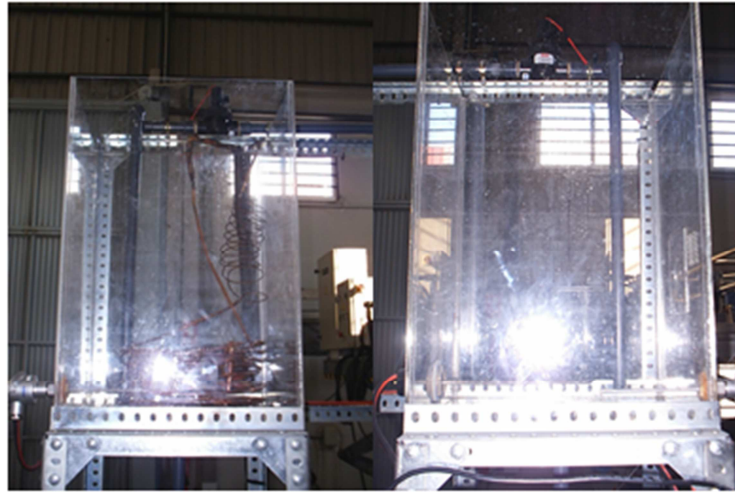


*Figura 8: Resistencia calefactora*

Estos depósitos tienen unas dimensiones de 0.4 x 0.4 x 0.6 m. con una capacidad de 80 litros cada uno. La diferencia entre ambos radica en el grosor de sus paredes, siendo de 4 mm en el depósito superior izquierdo y de 5 mm en el depósito superior derecho.

Por su parte, el depósito inferior tiene unas dimensiones de 0.45 x 0.45 x 0.80 y una capacidad de 160 litros. El grosor de sus paredes es también de 5 mm. En la *Figura 9 y 10* podemos ver los tres depósitos:





*Figura 9: Detalle depósitos superiores*



*Figura 10: Depósito central inferior*

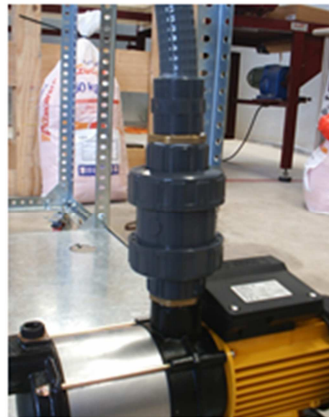
### La bomba.

Este motor se encarga de bombear el líquido desde el depósito inferior (mezcla) hacia los depósitos superiores, según la necesidad del sistema. La bomba utilizada, es una electrobomba centrífuga multietapa monobloque *ESPA* modelo *Prisma 15 2M* con potencia de 450 watios, como se muestra en la siguiente imagen (*Figura 11*).



*Figura 11: Vista de la bomba*

A la salida de la bomba se ha colocado una válvula anti-retorno para evitar la caída del líquido, como puede verse en la *Figura 12*:

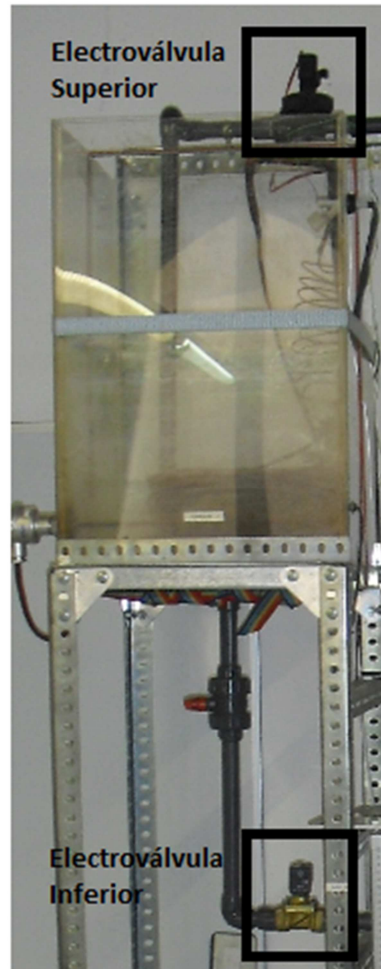


*Figura 12: Válvula anti-retorno*



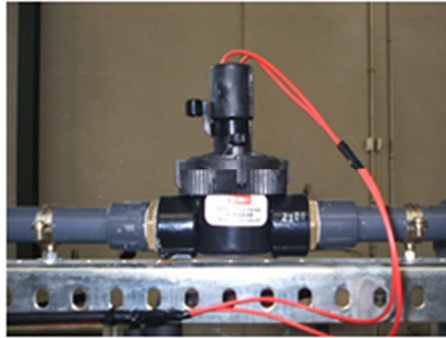
### Las electroválvulas.

Estos dispositivos se encargan de regular la entrada y salida de líquido en los depósitos superiores. Cada depósito contiene una válvula colocada en la parte superior que permite su llenado y otra para vaciarlo en la parte inferior como puede verse en la *Figura 13*:



*Figura 13: Detalle electroválvulas*

La electroválvula superior es una válvula *TORO* modelo *EZP-03-54* que puede trabajar con una presión desde los 0.7 hasta los 10 bares y está alimentada por una tensión de 24 Vac. a una frecuencia de 50 Hz. En la siguiente imagen (*Figura 14*) podemos observar la válvula descrita.



*Figura 14: Detalle electroválvula superior*

Por otro lado, la electroválvula inferior de la marca *ODE*, trabaja sin presión diferencial siendo alimentada por la misma tensión que la válvula superior. La *Figura 15* nos muestra la electroválvula inferior.



*Figura 15: Detalle electroválvula inferior*

Por último, y por motivos de seguridad, a la salida de cada depósito, se ha colocado una válvula manual de *PVC* como muestra la imagen siguiente (*Figura 16*):



*Figura 16: Detalle válvula manual*

### 3.2. Sensorización de la maqueta.

Los sensores son una pieza fundamental para dotar a la maqueta de “sentidos” que luego serán interpretados por los elementos de control del sistema. Podríamos decir que los sensores son a cualquier proceso industrial, lo que la vista, el oído, el tacto... a los seres humanos. Por lo tanto, de la misma forma que nuestro cerebro interpreta las señales que le llegan de nuestros sentidos y actúa en consecuencia, el dispositivo de control recibirá las señales de los sensores y dará una respuesta según las necesidades del proceso a controlar.

De esta forma, definimos al sensor como un dispositivo capaz de detectar y medir cualquier tipo de manifestación física como puede ser la humedad, la presión o el nivel. Además, algunos son capaces de aprovechar sus propias características físicas, como pudiera ser el termómetro de mercurio, que aprovecha las cualidades de este metal líquido para medir temperatura.

Estos transductores son capaces de proporcionar una señal, digital o analógica según lo cometido, adecuada para su recepción e interpretación a partir de la magnitud física que están midiendo. Existen varios tipos de sensores y muy variados (tantos como magnitudes) para medir un mismo tipo de manifestación física, por lo tanto, habrá que tener en cuenta, a la hora de elegir un sensor, qué tipo de sistema físico tenemos y las necesidades de control del mismo. En el caso particular de la maqueta de la que disponemos, necesitamos sólo dos tipos de sensores capaces de medir: nivel y temperatura.

Por lo tanto, para elegir los dispositivos correspondientes para la maqueta se ha tenido en cuenta si se pretendía obtener una lectura continua (señal de salida analógica) o simplemente indicar valores máximos y mínimos (señal de salida digital).

Como mostraba la *Figura 6* donde observábamos la disposición de los sensores en la maqueta, según las necesidades de la misma, en la siguiente tabla (*Figura 17*) se clasifican los sensores del proyecto según sus características.

Tipo de sensor	Magnitud	Tipo de salida	Alimentación
Interruptor-Flotador	Nivel máximo	Digital; Todo-Nada	/
Óptico	Nivel mínimo	Digital; Todo-Nada	12 Vdc
Sonda de nivel	Nivel	Analógica; Transmisor a dos hilos de 4 – 20 mA.	24 Vdc
PT100	Temperatura	Analógica; Transmisor a dos hilos de 4 – 20 mA.	24 Vdc
SITRANS Probe LU*. Inteligente por ultrasonidos.	Nivel, distancia y caudal	Digital; Transmisor a dos hilos de 4 – 20 mA para estándar HART. Convertidor A/D para bus de campo PROFIBUS.	Bus de campo PROFIBUS PA a 24 Vdc.

Figura 17: Tabla de sensores

SITRANS Probe LU\*: Explicado en el capítulo 5 de este proyecto.

### 3.2.1. Sensores de salida digital.

Como ya hemos apuntado anteriormente, podemos distinguir dos tipos de sensores según la salida entregada. Vamos a comenzar con los sensores de salida digital, que son los más sencillos utilizados en la maqueta.

#### **Interruptor flotador de nivel (nivel máximo).**

Estos sensores capacitivos se encuentran en los dos depósitos superiores de la maqueta y se encargarán de indicar cuando el tanque ha alcanzado su nivel máximo de llenado. Estos interruptores pueden trabajar en dos posiciones distintas: normalmente abierto y normalmente cerrado. En nuestro caso, están normalmente abiertos.

Consiste, básicamente, en dos láminas de metal finas que hacen de contactos y un imán actuador que se excitará según haya contacto entre ellas. De esta forma, quedan aisladas la parte de transmisión de señal y la parte de contacto.

Su funcionamiento es sencillo. En primer lugar hay que determinar el nivel máximo de llenado del depósito -en nuestro caso a unos 72 cm. desde la base del mismo- y colocar el sensor en ese punto. Hecho esto, cuando el tanque empiece a llenarse de líquido y este alcance al flotador del sensor, se cerrará el contacto y se activará el imán actuador, transmitiendo una señal eléctrica que indica que el nivel máximo se ha alcanzado. La imagen siguiente (*Figura 18*) muestra al sensor, normalmente abierto, instalado en el depósito.



*Figura 18: Sensor normalmente abierto*

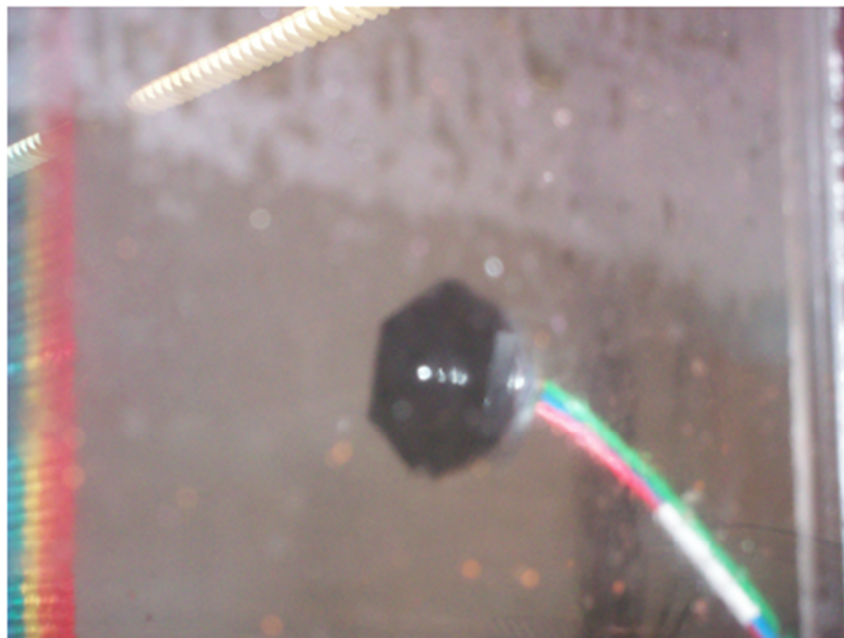
Características técnicas:

- Temperatura de funcionamiento: de  $-20$  a  $+75^{\circ}\text{C}$
- Tensión máxima: 240V ac./120V dc.
- Corriente máxima: 0.6 A.
- Vida útil eléctrica: 500000 funcionamientos.
- Paso de rosca: M16 por 2 por 19 mm.
- Peso específico mínimo del fluido: 0.85

**Sensor óptico de nivel (nivel mínimo).**

Si los sensores capacitivos se encargaban de indicar cuándo se alcanzaba el nivel máximo de llenado de los depósitos superiores, los sensores ópticos se encargan justamente de lo contrario. Una vez que se haya determinado el nivel mínimo -unos 8 cm en nuestro caso desde la base del depósito- se colocarán en su lugar correspondiente.

Basándose en el principio de reflexión total, estos sensores utilizan un led y un fotosensor integrados, de forma que, cuando el líquido no cubre al sensor, se establece un camino óptico entre ellos, indicando así que se ha alcanzado el nivel mínimo de llenado del tanque. En la *Figura 19* observamos el sensor óptico colado en uno de los depósitos superiores.



*Figura 19: Sensor óptico*

Características técnicas:

- Tensión de alimentación: 5 a 12V dc.
- Temperatura de funcionamiento: de  $-25$  a  $+80^{\circ}\text{C}$
- Corriente de led nominal: 15mA.
- Corriente de sumidero (máx): 10mA a  $25^{\circ}\text{C}$
- Rosca de montaje: M12 por 1mm.
- Consistencia dispositivo-dispositivo:  $\pm 1$ mm.
- Histéresis: 2mm.

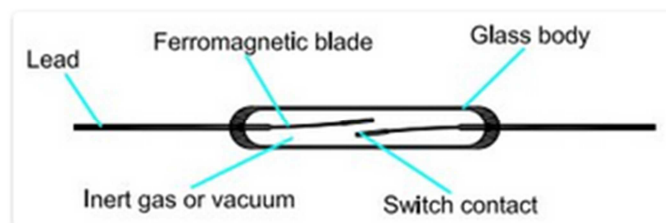
### 3.2.2. Sensores de salida analógica.

Una vez que conocemos los sensores de salida digital que actúan en nuestro sistema físico, el siguiente paso será entender los sensores de salida analógica, que son parte fundamental del proceso.

#### Sonda de nivel analógico (nivel).

La sonda de nivel analógica se encuentra ubicada en el depósito inferior centra (tanque de mezcla) y está compuesta por un tubo de medición con un imán incorporado al flotador del mismo y un transmisor que enviará la señal al controlador lógico.

En el interior del tubo de medición se encuentran distintos interruptores *reed*. Estos dispositivos están formados por dos laminas ferromagnéticas, generalmente compuestas de *Ni* y *Fe*, herméticamente selladas en una cápsula de vidrio. Las hojas se superponen internamente en la cápsula de vidrio dejando solo un pequeño espacio entre ellas, y se ponen en contacto ante la presencia de un campo magnético adecuado. El área de contacto de ambas hojas se recubre con metales duros (Rodio o Rutenio) que elevan la vida útil de los contactos. En la imagen siguiente (*Figura 20*) podemos ver un interruptor *reed*.



*Figura 20: Interruptor reed*

Cada interruptor lleva también una resistencia que hará variar la tensión que llega al transmisor. Por lo tanto, el funcionamiento es simple. Según vaya subiendo el nivel del

depósito, el flotador con su imán irá deslizándose por el tubo de medición cerrando los contactos de los interruptores *reed*. De esta forma, el voltaje que llega al transmisor a través del tubo de medición va variando según el nivel del líquido.

El transmisor se puede montar en la caja terminal o como instrumento externo. La indicación analógica o digital local también se puede proporcionar para satisfacer los requisitos para mayores precisiones de medición y monitorización. En la *Figura 21* se muestra el transmisor montado en la caja terminal para su funcionamiento junto al tubo de medición.



*Figura 21: Sonda de nivel analógica*

Debemos tener en cuenta una serie de consideraciones a la hora de utilizar esta sonda o sensor:

- Diseñado para automatización de llenado o vaciado en pequeños tanques conteniendo agua, soluciones acuosas, químicos, aceites o líquidos de baja viscosidad en general
- Permiten obtener contactos en distintos niveles, señales de control o de mando en cualquier punto del recorrido del flotador.
- Sistema de acople magnético, garantiza total aislamiento del circuito eléctrico.
- Se proveen ejecuciones especiales de acuerdo a las necesidades particulares de cada industria.

En las *Figuras 22 y 23* se muestran respectivamente, una imagen del sensor con sus medidas y acotaciones y el esquema de conexionado del transmisor.

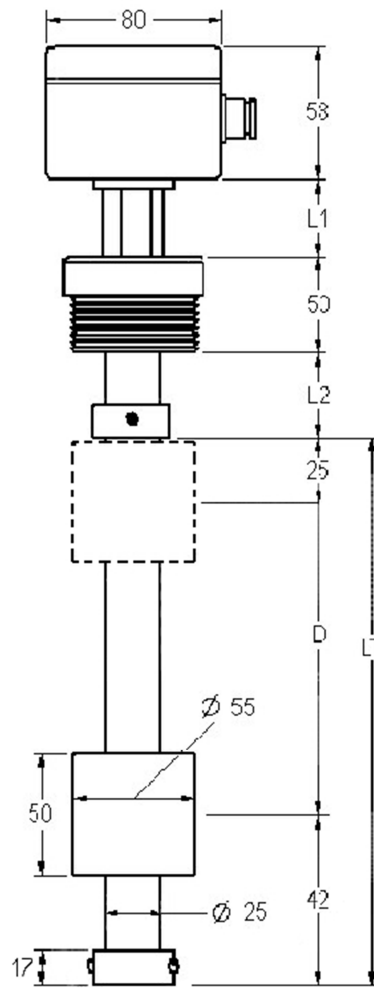


Figura 22: Medidas del sensor

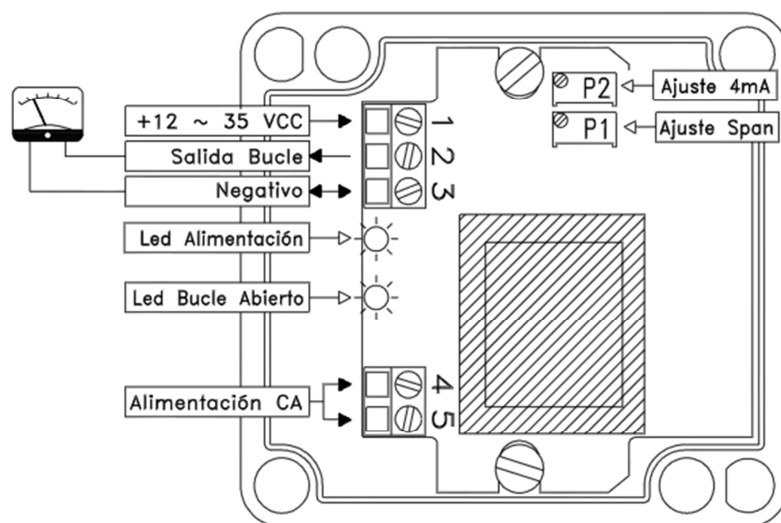


Figura 23: Esquema de conexionado



Características técnicas:

- Caja de conexiones en policarbonato 80x85x55mm.
- Separador (L1) estándar de 13 mm.
- Conexión al proceso por medio de tapón roscado de PVC 2”1/2 G.
- Tubo de guía de 25 mm de diámetro de PVC.
- Mínima distancia de actuación (L3):
 

Paso 5:	30 mm.
Paso 10:	35 mm.
- Tope inferior de PVC.
- Flotador cilíndrico de 55 mm de diámetro x 50 mm de PVC.
- Temperatura soportada:
 

En aire:	-20 a +50 °C.
En agua:	-20 a +60 °C.
- Clase de protección IP 65.
- Tensiones de alimentación:
 

12-35	Vdc.	(Bornes 1-3)
24	Vac.	(Bornes 4-5)
48	Vac.	(Bornes 4-5)
110	Vac.	(Bornes 4-5)
230	Vac.	(Bornes 4-5)
- Distancia entre contactos estándar de 10 mm.
- Transmisor a dos hilos con señal de salida de 4 a 20 mA.
- Longitud (D) de 150 a 2500 mm.
- Repetibilidad: +/- 1%

Teniendo en cuenta las características del sensor, y sabiendo que está alimentado con 24 Vdc. (*Figura 17*), podremos calcular su resistencia máxima de carga del bucle de corriente según nos indica el fabricante en la siguiente gráfica (*Figura 24*):

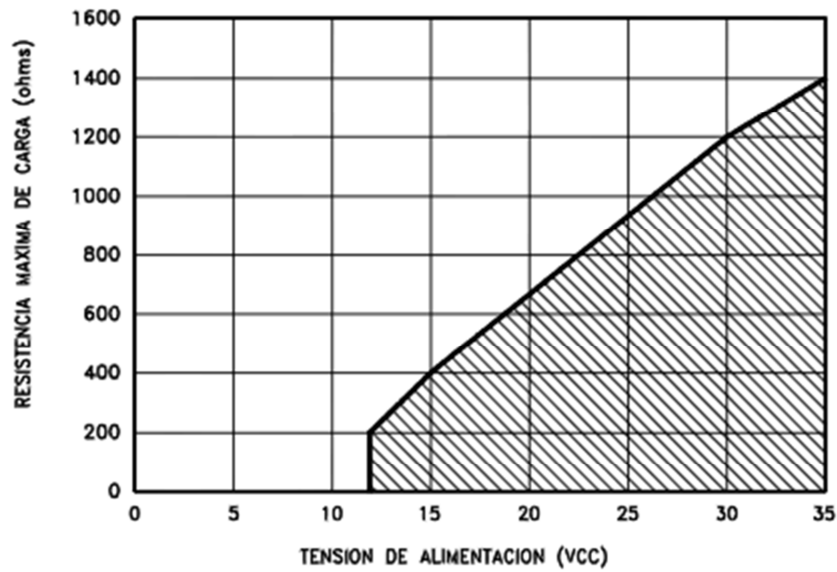


Figura 24: Resistencia máxima de carga

Para la alimentación en corriente alterna (bornes 4 y 5 del esquema de conexionado) la resistencia de carga no deberá exceder de 800 ohmios.

Por su parte, si queremos obtener la resolución del sensor, deberemos aplicar los cálculos que se muestran a continuación (Figura 25), obteniendo los parámetros pertinentes.

L : Altura total (mm)  $L = D + 30$  (35)  
 D : Distancia a controlar (mm)  
 N : Número de lecturas  $N = D / P$   
 P : Paso entre lecturas (mm)  
 R : Resolución (mA/lectura)  $R = 16 / N$

D \ P	P	
	5	10
150	0,53	1,06
500	0,16	0,32
1000	0,08	0,16
1500	0,05	0,11
2000	0,04	0,08
2500	0,03	0,06

Figura 25: Cálculo de parámetros y resolución

### Cálculo de la resolución

El cálculo de la resolución resultará sencillo siguiendo las ecuaciones anteriores. El primer paso será obtener la distancia a controlar,  $D$ . Como ya hemos comentado al explicar los sensores de salida digital, nuestro nivel máximo se encontrará a 72 cm desde la base del depósito, mientras que nuestro nivel mínimo estará a 8 cm de dicha base. Por lo tanto la distancia a controlar será:

$$D = 72 - 8; D = 64 \text{ cm o } 640 \text{ mm.}$$

El siguiente paso será calcular el número de lecturas,  $N$ . Para ello necesitamos el paso entre lecturas, que en nuestro sensor, es de 10 mm. Por lo tanto,  $N$  valdrá:

$$N = 640/10; N = 64$$

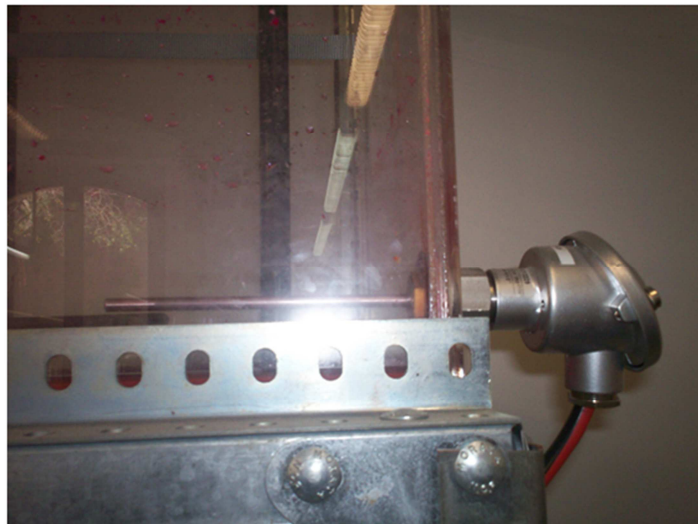
A continuación, calculamos la resolución  $R$  aplicando la ecuación correspondiente de la *Figura 25*:

$$R = 16/N; R = 16/64; R = 0.25 \text{ mA/lectura}$$

Esto quiere decir que, si la lectura del sensor es 4 mA cuando el nivel del líquido se encuentra en el punto mínimo (8 cm de la base del depósito), su valor irá incrementando en 0.25 mA por cada centímetro que aumente el nivel del líquido hasta llegar al punto máximo (72 cm de la base del depósito) que corresponderá con 20 mA.

### **Sensor de temperatura Pt100.**

Para medir la temperatura de los depósitos, se ha utilizado una sonda *Pt100* de la marca *JUMO*. Son detectores resistivos de temperatura, *RTD*, que basan su funcionamiento en el aumento de su resistencia conforme aumenta la temperatura. Al estar hecha de platino (*Platinum Resistance Thermometer, PRT*), a una temperatura de 0°C su valor resistivo será de 100 ohmios. Esta es una característica común a todas las *Pt100*. Dicha sonda incorpora un transmisor modelo *dTrans T03 B*, también de la marca *JUMO*. En la *Figura 26* podemos ver la resistencia instalada en el depósito superior derecho y su transmisor en el cabezal de la misma.



*Figura 26: Sonda de temperatura Pt100*

Si queremos entender el funcionamiento de este sensor, debemos tener en cuenta el comportamiento del material a nivel atómico. Al aumentar la temperatura, los átomos comienzan a vibrar haciendo que los electrones choquen entre sí dificultando el paso de corriente y aumentando el valor resistivo del dispositivo. De esta forma, se dice que la *Pt100* posee un coeficiente de temperatura positivo.

Teniendo en cuenta lo expuesto anteriormente, su ecuación característica queda de la siguiente forma:

$$R_T = R_0 (1 + \alpha T);$$

siendo

- $R_T$ : El valor de la resistencia para una determinada temperatura.
- $R_0$ : El valor de la resistencia a 0°C, de 100 ohmios.
- $\alpha$ : El coeficiente de temperatura de la sonda.
- $T$ : La temperatura a la que se quiere conocer el valor resistivo del dispositivo.

A la hora de trabajar con las sondas *Pt100*, debemos tener en cuenta las siguientes consideraciones:

- No se podrán medir temperaturas próximas al punto de fusión del conductor.
- Se evitará el auto-calentamiento provocados por el circuito de medida. Para minimizar este error por auto-calentamiento se limita la corriente que introduce en el sensor el circuito de acondicionamiento.
- Posibilidad de deformaciones mecánicas, es decir, la variación de resistencia se podría deber tanto al cambio de resistividad como al cambio de dimensiones asociado al incremento de temperatura.

Estos requisitos no nos afectarán puesto que el fabricante de sondas de temperatura ya los habrá tenido en cuenta a la hora de su creación. Por otro lado, podemos destacar como ventajas:

- Sensibilidad diez veces mayor que los termopares.
- Alta repetición, estabilidad y exactitud en la medida.
- Gran margen de linealidad, prácticamente lineal en todo su rango de funcionamiento.

Por su lado, las características del sensor y de su transmisor incorporado vendrán dadas según el modelo utilizado dentro de la gama de sondas *JUMO*. En la *Figura 27* podemos observar el sensor dividido en sus dos partes importantes: sensor y cabezal en el que se encuentra su transmisor.

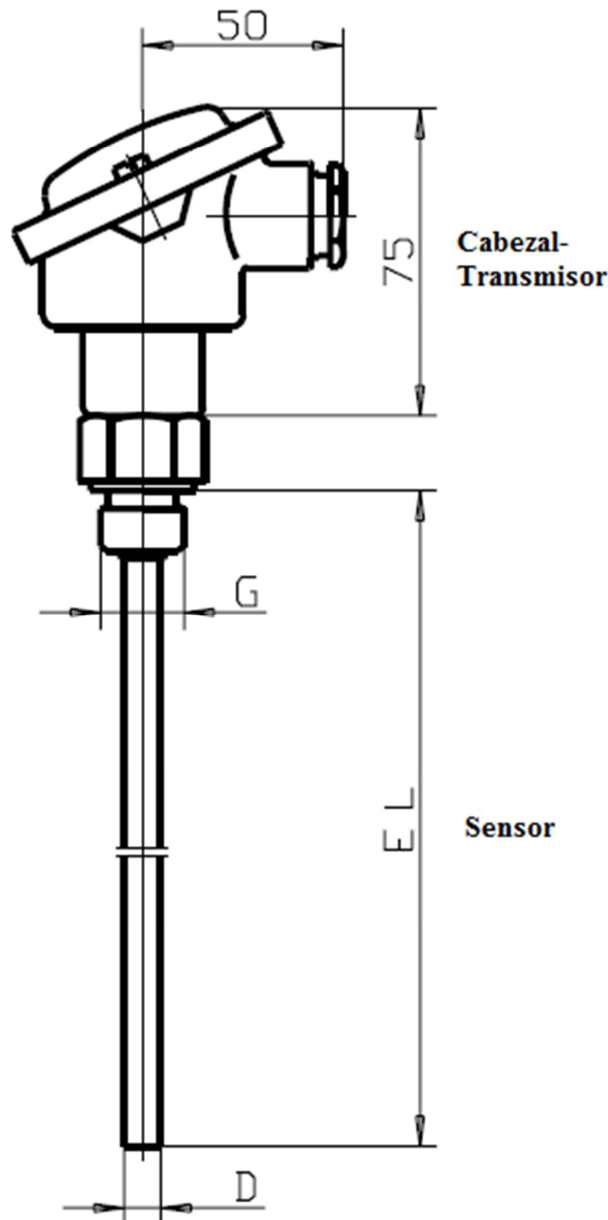


Figura 27: Pt100

### Características del sensor de platino

El sensor fabricado en platino pertenece a la gama *Type 902002*, siendo el modelo utilizado el *Type 902002/20*. Estos termómetros de termistancia de montaje por rosca se utilizan fundamentalmente para la medición de temperaturas en medios líquidos o gaseosos. La hermeticidad de máxima eficacia de esta forma de montaje, tanto para baja presión como en vacío, es un criterio decisivo para su selección. Los campos de aplicación más importantes se refiere a las técnicas de climatización y refrigeración, fabricación de equipos de calefacción, hornos, construcción de maquinaria y aparatos de medida, así como para la industria química.

La *Pt100* tiene la ventaja que es altamente resistente a la corrosión, es relativamente fácil trabajar, está constituida con un estado muy puro del platino y por esto exhibe una buena repetibilidad de sus características eléctricas. Posee una gran sensibilidad, puesto que la recta de la curva de características (*Figura 28*) es directamente proporcional al valor de  $R_0$ , el valor nominal. Una de sus mayores ventajas, es el hecho de que su resistencia tiene un

cambio muy grande con la temperatura. El cambio de la resistencia en la gama de éste modelo, con temperaturas inferiores a 100°C es aproximadamente de 0.4 ohmios/°C.

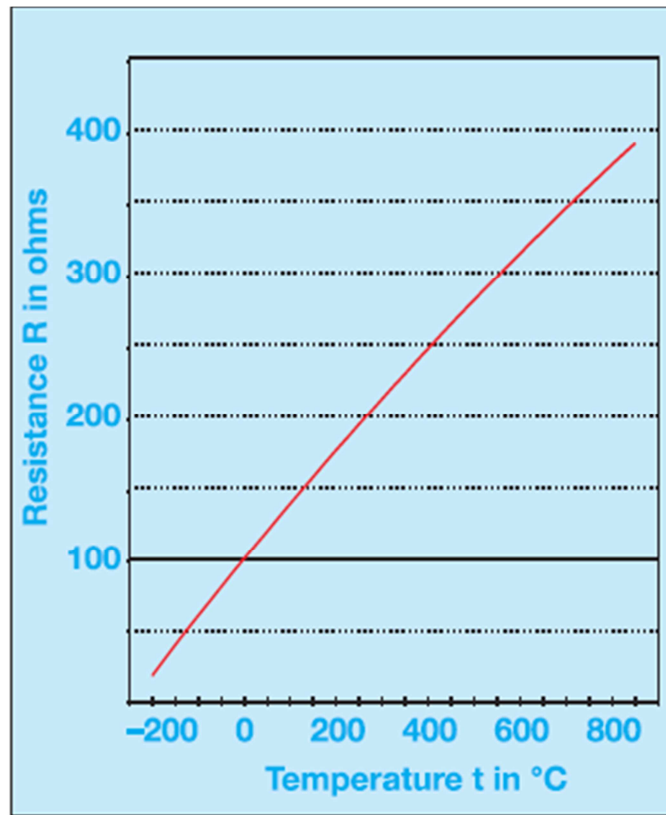


Figura 28: Curva característica de la Pt100

Como podemos observar en la curva anterior, la variación de la resistencia con el incremento de la temperatura sigue una progresión prácticamente lineal, como ya hemos apuntado con anterioridad.

Como parámetro adicional, el estándar define el coeficiente de temperatura,  $\alpha$ , entre 0 y 100 °C. Este parámetro representa el cambio medio en la resistencia, referida al valor nominal entre  $R_0$  y  $R_{100}$ , cuyos valores de resistencia son tomados para las temperaturas 0 y 100°C respectivamente.

$$\alpha = \frac{R_{100} - R_0}{R_0 \times 100^\circ\text{C}} = 3.850 \times 10^{-3} \text{ } ^\circ\text{C}^{-1}$$

Respecto a la tolerancia de la Pt100, el fabricante ha definido las clases de la tolerancia, basada en las tolerancias estándar mostradas en las Figuras 29 y 30, para cubrir la gama de usos posibles más grande a través del mercado.

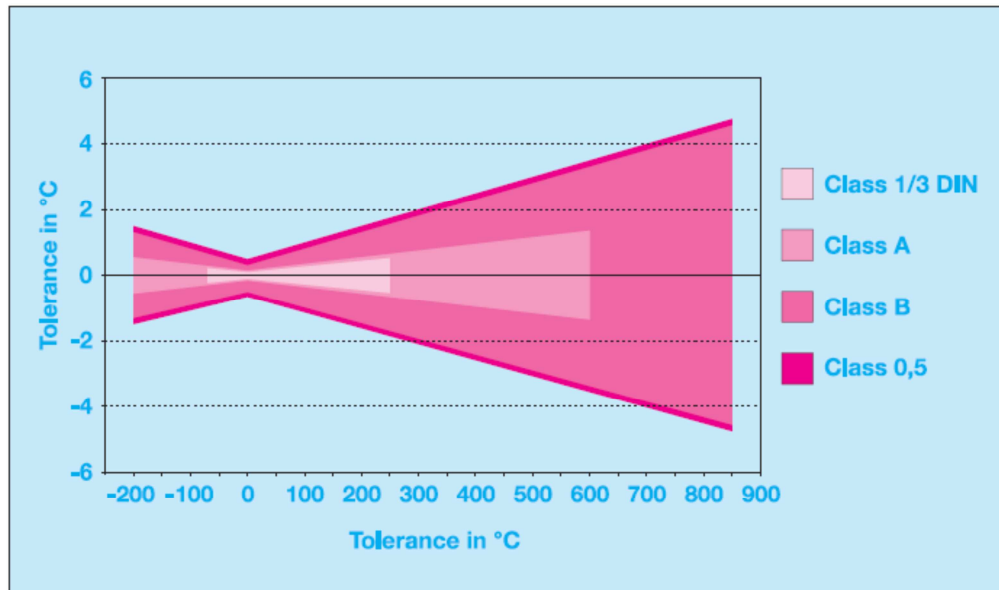


Figura 29: Clases de tolerancia

Tolerance class	Sensor category	Temperature range	Tolerance in °C
Class 1/3 DIN B	Thin-film Wire	-50 to +200 °C -70 to +250 °C	$\pm (0.10K + 0.0017 \times ItI)$
Class A	Thin-film Wire	-70 to +300 °C -200 to +600 °C	$\pm (0.15K + 0.002 \times ItI)$
Class B	Thin-film Wire	-70 to +600 °C -200 to +850 °C	$\pm (0.30K + 0.005 \times ItI)$
Class 0.5	Thin-film Wire	-70 to +600 °C -200 to +850 °C	$\pm (0.50K + 0.006 \times ItI)$

Figura 30: Tolerancias estándar

En cuanto a la respuesta de la *Pt100*, si el sensor de temperatura sufre un cambio repentino de temperatura, sufrirá un retraso de tiempo antes de que haya tomado la nueva temperatura. Este retraso dependerá mucho del sensor de temperatura y de las condiciones ambientales. Las figuras dadas en las hojas de datos (ver anexos de sensores) están referidas a medidas en agua agitada, en un flujo de  $v = 0.4$  metro/sec o en aire en un flujo de 1 metro/sec. Los tiempos de reacción para otros medios se pueden calcular con la ayuda del coeficiente térmico según *VDI/VDE 3522*. La siguiente imagen (*Figura 31*) nos muestra una curva de respuesta típica (función de la transferencia) a un cambio repentino de temperatura.

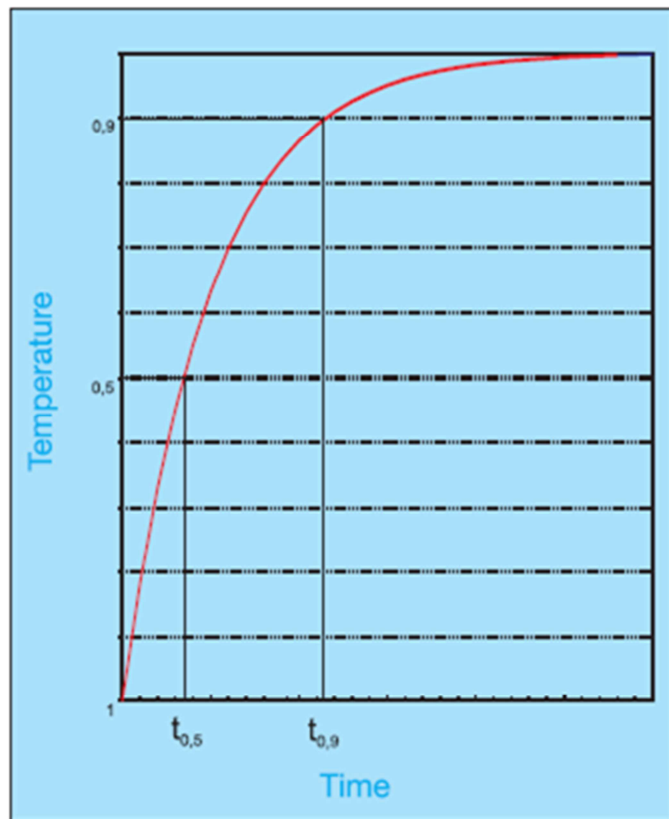


Figura 31: Curva de respuesta a incremento de temperatura

Ahora que ya conocemos las características técnicas y de funcionamiento del sensor de platino, el siguiente paso será hacer lo propio con el transmisor que incorpora el sensor. Como ya hemos citado anteriormente, el transmisor pertenece al modelo *dTrans T03 B* con conexión a dos/tres hilos.

Características técnicas del transmisor *dTrans T03 B*:

- Rango de medida: -200 a 850 °C
- Circuito de conexión: 2/3 hilos.
- Span mínimo: 25 °C
- Span máximo: 1050 °C
- Corriente del sensor:  $\leq 0.5$  mA
- Velocidad de muestreo: continua
- Calibración: en °C o °F. Rangos ajustables. De -50 a 400 °C en nuestro caso.
- Rotura de sonda y cables:
 

por encima	$\geq 22$ mA a $< 28$ mA
por debajo	$\leq 3.6$ mA



- Salida: 4 – 20 mA
- Características de transferencia: lineal con la temperatura
- Precisión:  $\leq \pm 0.1\%$
- Resistencia de carga ( $R_b$ ):  $R_b = (U_b - 7.5 \text{ V}) / 22 \text{ mA}$
- Alimentación: 7.5 a 30 V
- Rango de temperatura de funcionamiento del transmisor: -40 a 85 °C

Dentro de las características técnicas arriba citadas, debemos tener cuidado con el span. El span nos indicará el rango disponible en el que el sensor podrá trabajar. La siguiente gráfica (Figura 32) muestra el span permitido del dispositivo.

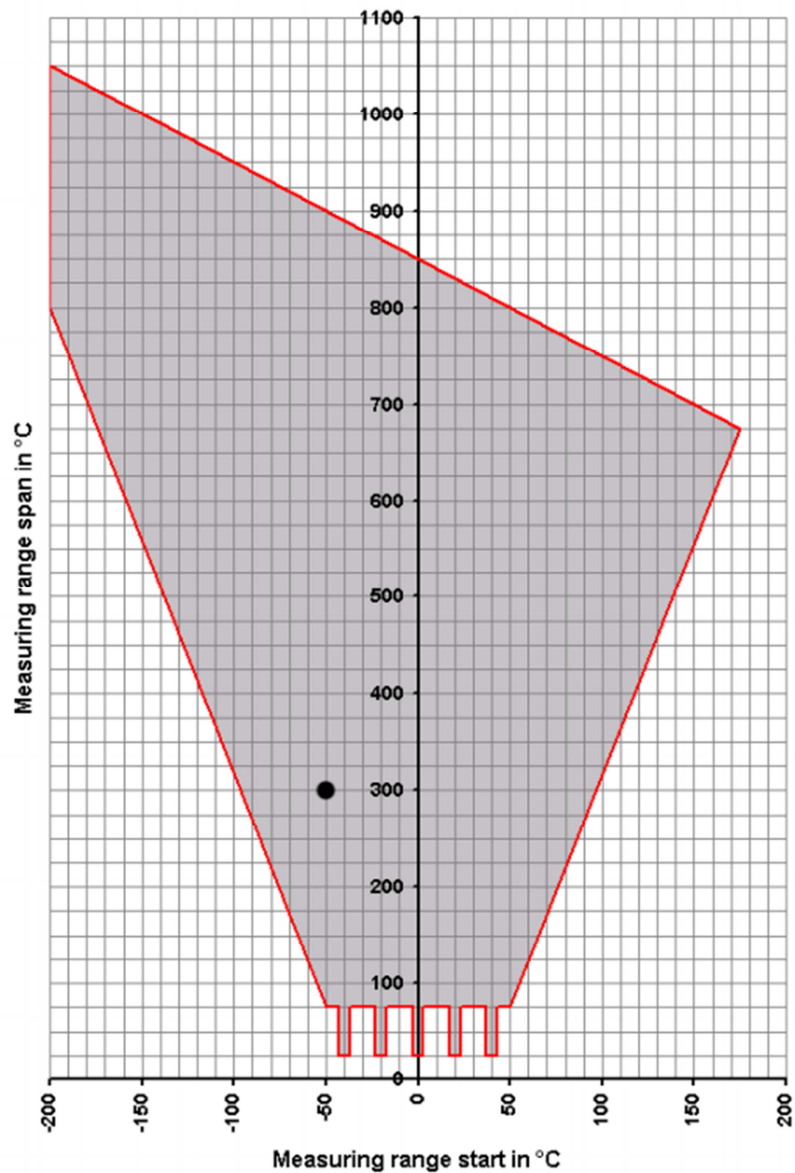
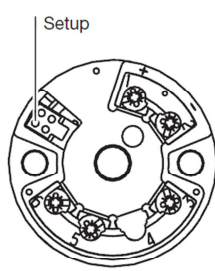








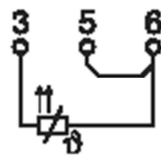
Figura 32: Rango de span permitido

El rango del span se calculará de forma sencilla. Si tenemos un rango de temperaturas desde  $-50$  hasta  $250\text{ }^{\circ}\text{C}$ , el rango del span será la resta del rango final y el rango inicial. Por lo tanto:

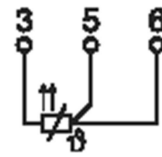
Rango de span:  $250 - (-50) = 300$ ; Ver *Figura 32*.

Por último, también es importante saber cómo conectar y cablear el dispositivo. En la *Figura 33* se muestra como hacer una conexión a dos o tres hilos mostrando el cabezal del transmisor y sus bornes de conexión.

	Connection for		Terminal assignments		
		Supply voltage 7.5 – 30V DC	+1	$R_B = \frac{U_b - 7.5V}{22mA}$	1 2
		Current output 4 – 20mA	-2	$R_B$ = burden resistance $U_b$ = supply voltage	+ -
	Analog inputs				
		Resistance thermometer in 2-wire circuit	3 5 6	standard is $R_L = 0\Omega$	
		Resistance thermometer in 3-wire circuit	3 5 6	$R_L < 11\Omega$ $R_L$ = lead resistance per conductor	



Sonda a dos hilos



Sonda a tres hilos

*Figura 33: Conexión a dos y tres hilos del sensor*

\*Consultar los anexos de sensores para más detalle de los mismos.

### 3.3. Instalación eléctrica de la maqueta.

Como ya se ha citado con anterioridad, todo el cuadro eléctrico de la maqueta se encuentra ubicado dentro de un armario. Además, en la puerta del armario se puede ver una pantalla táctil para controlar la maqueta de forma manual y el interruptor de apagado/encendido del cuadro tal y como muestra la *Figura 34*:



*Figura 34: Pantalla táctil e interruptor de encendido*

En la imagen a continuación (*Figura 35*) podemos ver el cuadro eléctrico con la parte posterior de la puerta del armario donde se encuentran las conexiones de la pantalla táctil.



*Figura 35: Detalle del cuadro y la pantalla táctil*

#### 3.3.1. El cuadro eléctrico.

Si observamos con detenimiento la *Figura 36* podemos ver que el cuadro eléctrico está dividido en tres partes claramente diferenciables entre sí.

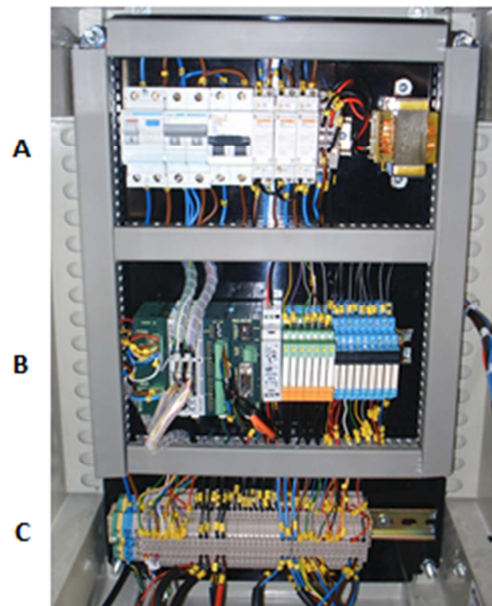


Figura 36: Cuadro eléctrico

A) En la parte superior se encuentra todos los elementos que componen el esquema de fuerza de la maqueta, que pueden observarse en la *Figura 37*:

1. Un interruptor magneto-térmico.
2. Un interruptor diferencial de 16 Amperios.
3. Un interruptor diferencial de 6 Amperios.
4. Tres contactores (*KM 1*, *KM 2* y *KM 3*).
5. Un transformador de 220/24 V/AC

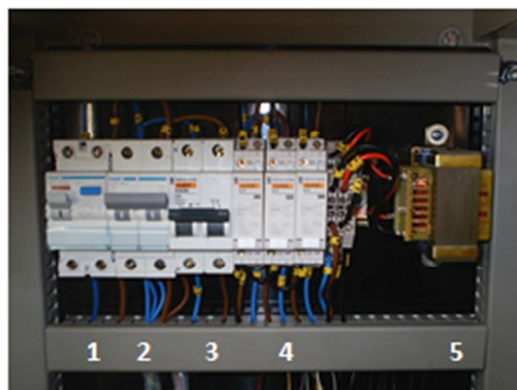
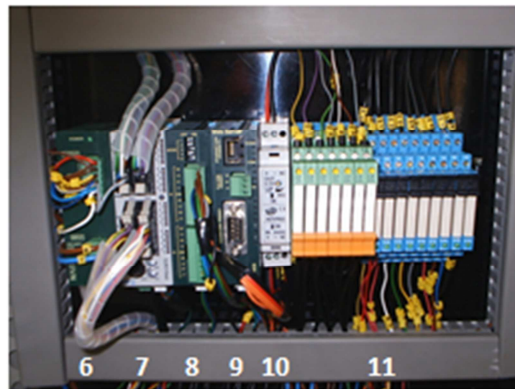


Figura 37: Elementos del esquema de fuerza

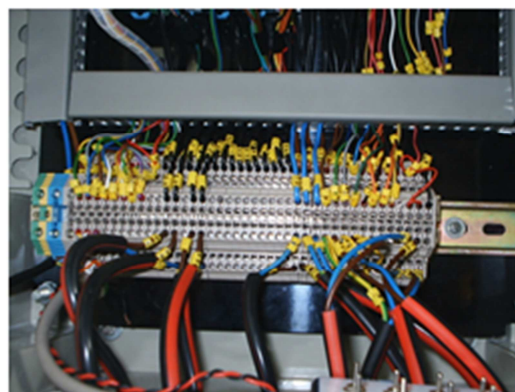
**B)** En la parte central se encuentra la zona de actuadores y controladores que se compone de los siguientes elementos que se muestran en la *Figura 38*:

6. Una fuente de alimentación de 24 V dc *FP0-PSA2*.
7. Un *PLC* unidad de control *Nais FPΣ C32T2*.
8. Un módulo de entradas analógicas *A/D FP0A80*.
9. Un módulo *Web Server FP-Web*.
10. Un convertidor de 24Vcc/5Vcc *ADVR02*.
11. Dieciséis módulos de relés (desde Relé 0 a Relé F).



*Figura 38: Actuadores y controladores*

**C)** En la parte inferior encontramos las bornas de conexión. De ellas parten y llegan todo el cableado de la maqueta. Tienen dos conexiones por borna y son de prensa fácil. Hay que destacar también que la numeración es la misma que podemos encontrar en los planos y anexos. En la imagen a continuación (*Figura 39*) se muestra la última parte del cuadro eléctrico.



*Figura 39: Conexiones*

### 3.3.2. Esquemas eléctricos.

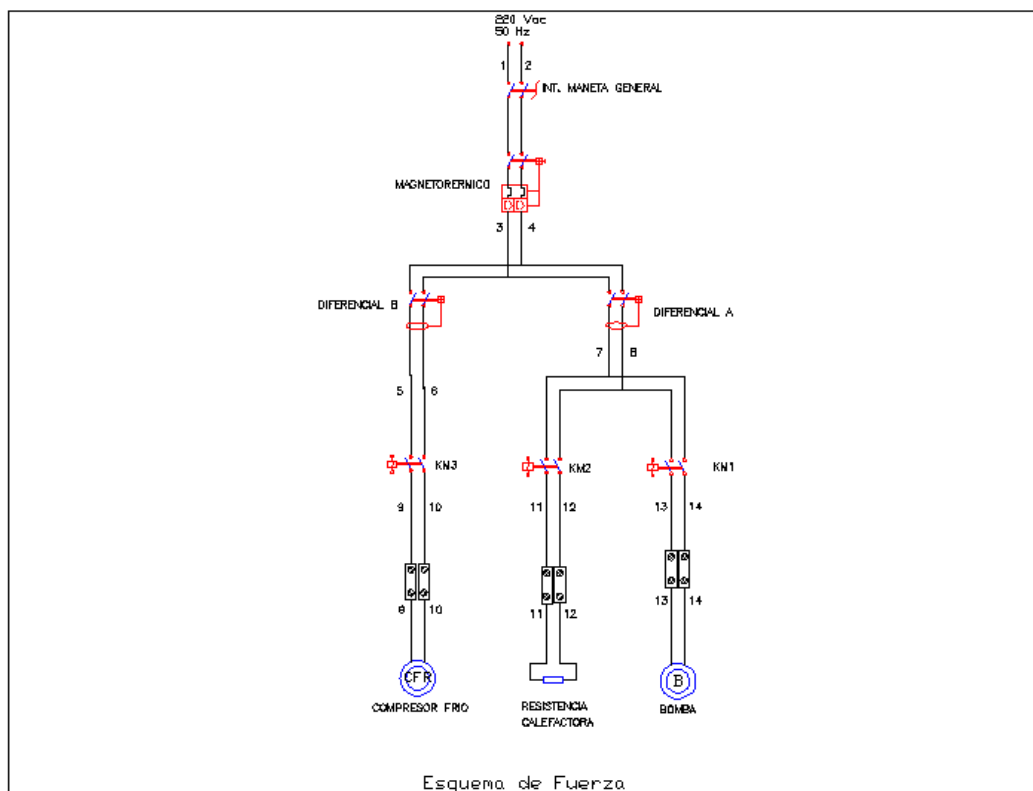
Una vez que ya sabemos cómo se distribuye el cuadro eléctrico, el siguiente paso será conocer los esquemas eléctricos implementados en dicho cuadro.

#### Esquema de fuerza.

En la *Figura 40* podemos ver el esquema eléctrico correspondiente a los dispositivos que más consumo de potencia tienen. Las potencias consumidas se relatan a continuación:

- Máquina compresora de frío: 1500 watios.
- Resistencia calefactora: 750 watios.
- Bomba de agua: 450 watios.

Como se aprecia en el plano, los elementos que forman dicho esquema están presentes en la zona superior del cuadro eléctrico.



*Figura 40: Esquema de fuerza*

Para el encendido general se ha instalado un interruptor de maneta que hará las veces de interruptor general del sistema cortando la alimentación de la tensión de entrada de red de 220 Vac.

Se ha instalado también como medida de seguridad un interruptor magnetotérmico marca *HAGER* modelo *CDC 728M* de 25 amperios con una sensibilidad de 30 mA, así como dos interruptores diferenciales del tipo:

- Diferencial A: marca *HAGER* modelo *MUN 516A* de 16 amperios.
- Diferencial B: marca *MERLIN GERIN* modelo *K60N* de 6 amperios.

Para el control del encendido y apagado de los elementos a controlar (compresor frío, resistencia calefactora y bomba de retorno de agua) se han instalado tres contactores marca *MERLIN GERIN* modelo *CT* de 16 amperios.

### Esquema de entradas analógicas.

En el plano siguiente (*Figura 41*) se muestra el esquema de las entradas y salidas analógicas de la maqueta así como el módulo *A/D FP0A80*. Se ha utilizado como fuente de alimentación auxiliar el módulo *FP PSA2* con un voltaje nominal de 24 Vdc.

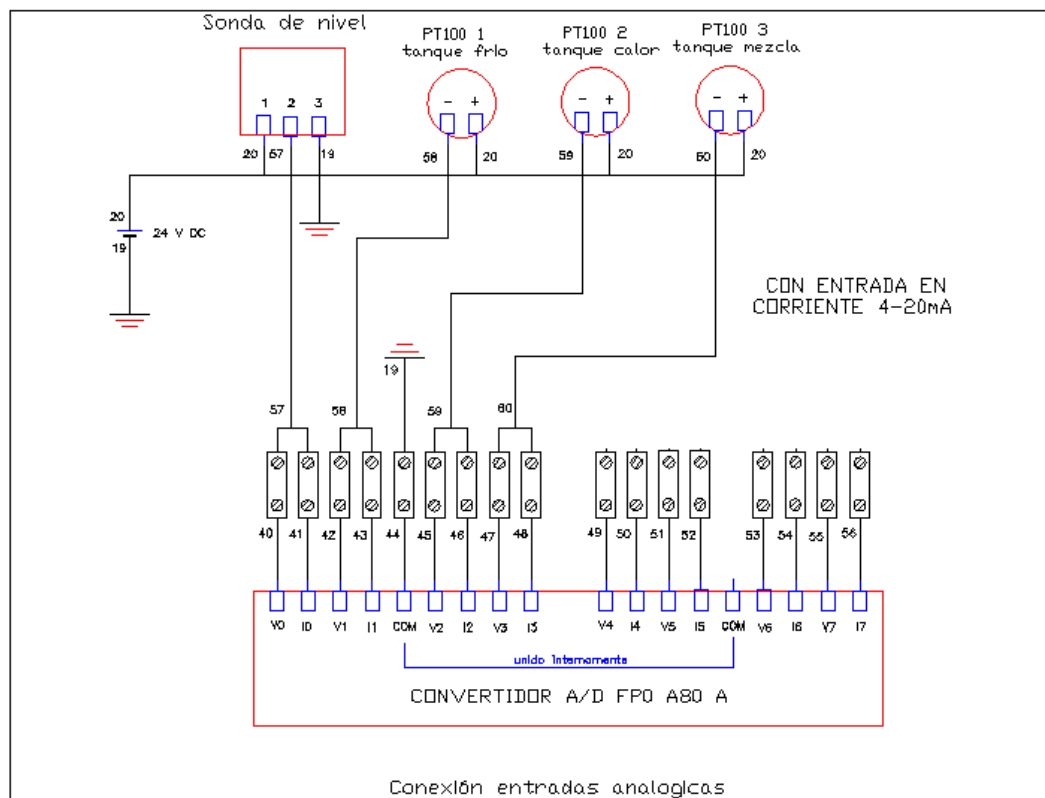


Figura 41: Esquema entradas analógicas de la maqueta

### 3.4. El autómatas de la maqueta. Unidad de Nais FPΣ C32T2.

La maqueta incorpora un pequeño controlador que se ocupa de las tareas propias de estos dispositivos. Hay que destacar que para la realización de este proyecto, este autómatas se encuentra desactivado. A continuación se explican las características de la unidad *Nais FPΣ C32T2*.

El FPΣ (Sigma) es un autómatas de reducido tamaño y con una gran funcionalidad. Permite la programación mediante instrucciones de estándar *IEC 61131-3*. Aunque sea un autómatas de tamaño reducido, presenta todas las funciones disponibles en los autómatas de gama media. Es capaz de almacenar 12 k pasos de programa así que no es necesario preocuparse por la memoria disponible a la hora de programar. Además dispone de 32.765 registros de datos por lo que puede manejar y almacenar una gran cantidad de información. En la imagen a continuación (*Figura 42*) se muestra el autómatas.



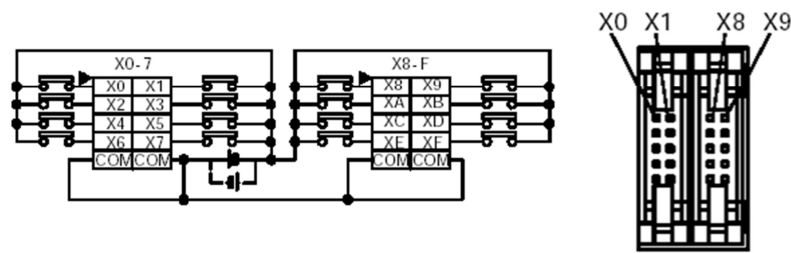
*Figura 42: Autómatas FPΣ*

#### 3.4.1. Entradas y salidas del autómatas.

El autómatas *FPΣ C32T2* dispone de un total de 16 entradas correspondientes desde la *X0* hasta la *XF* y 16 salidas correspondientes desde la *Y0* hasta la *YF*. La disposición de los pines se muestra en la *Figura 43*.



### Entradas



### Salidas

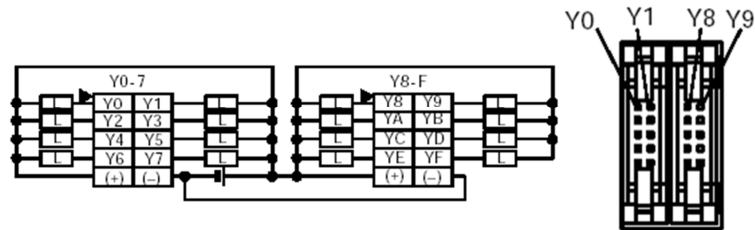
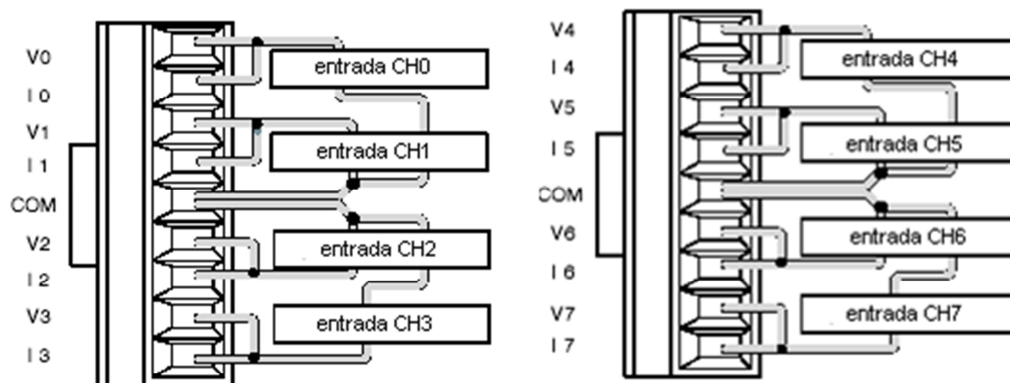


Figura 43: Distribución de las entradas y salidas del autómata

Los dos terminales (+) y (-) así como los cuatro (COM) están conectados internamente como se ve en la imagen anterior.

### 3.4.2. Módulo de entradas analógicas.

Para el control de la maqueta en el anterior proyecto, se dispuso de un convertidor Analógico/Digital modelo *FP0A80* provisto de 8 entradas analógicas. Dicho convertidor admite entradas analógicas en tensión y en corriente, adaptándose a los dispositivos que dispone la maqueta que proporcionan una señal de salida en corriente de 4 a 20 mA. La disposición de los pines del convertidor A/D se observa en la imagen siguiente (Figura 44):



**Disposición de entrada en corriente**

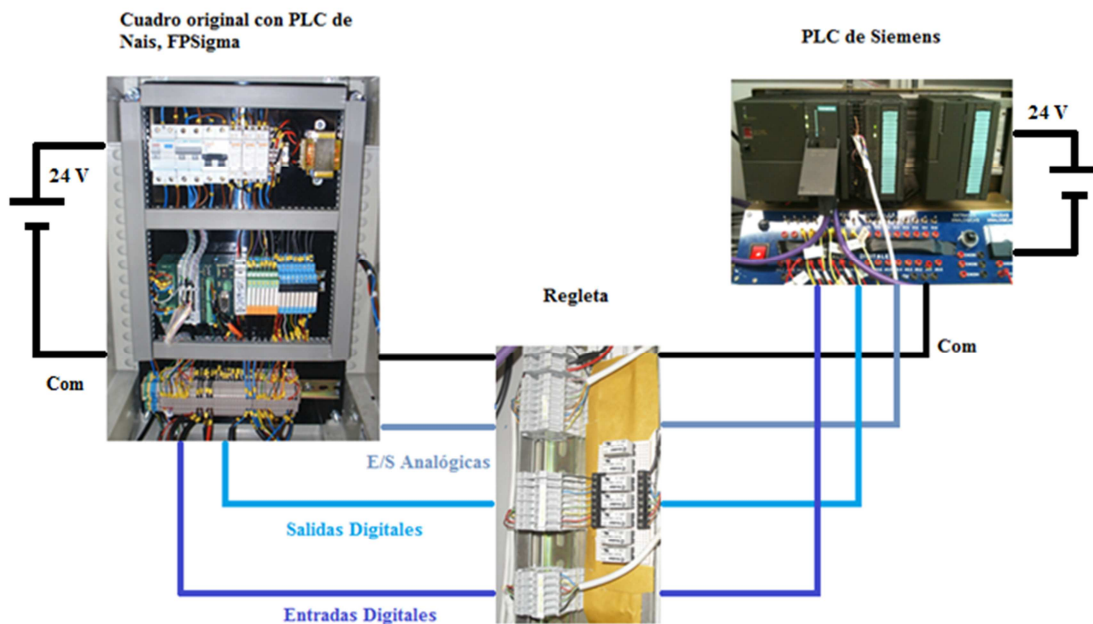
**Los terminales COM están unidos**

Figura 44: Disposición de los pines del convertidor analógico-digital

### 3.5. Modificaciones de la maqueta. Banco de pruebas Siemens.

Una vez que ya conocemos en profundidad la maqueta y los elementos de los que consta, el siguiente paso será optimizarla para el desarrollo de nuestro sistema de control. Para ello, se han llevado a cabo algunas modificaciones, como ya apuntáramos en la introducción de este capítulo, que permitirán la incorporación de un sistema esclavo-maestro compuesto por un autómatas y el sensor inteligente *SITRANS Probe LU*. Todo ello comunicado con un bus de campo *PROFIBUS*.

Para no perder el cuadro original, explicado anteriormente, se han duplicado las Entradas/Salidas del mismo, llevándolas a otro cuadro para poder conectarlas con el nuevo autómatas de *Siemens* tal y como muestra a continuación la *Figura 45*:



*Figura 45: Duplicado de E/S*

En la imagen anterior puede verse que se han utilizado una serie de relés para aislar las alimentaciones del *PLC* de *Siemens* con la del cuadro original en las salidas digitales.

Por su parte, las entradas digitales se han duplicado directamente (excepto los sensores ópticos) del cuadro original y se han conectado al entrenador del autómatas de Siemens. El entrenador está cableado internamente con el módulo de Entradas/Salidas digitales del *PLC*. Para conectar los sensores ópticos de los depósitos, ha sido necesario acondicionar la señal de los mismos, ya que ésta oscila entre los 5 y 12 voltios en continua de salida como hemos visto anteriormente. Las entradas digitales para el autómatas requieren de 24 Vdc. Para ello se ha dispuesto de una pequeña placa con un transistor y dos resistencias por cada sensor óptico para así obtener los 24 voltios necesarios. En la *Figura 46* podemos ver dicha placa integrada en el cuadro eléctrico original.

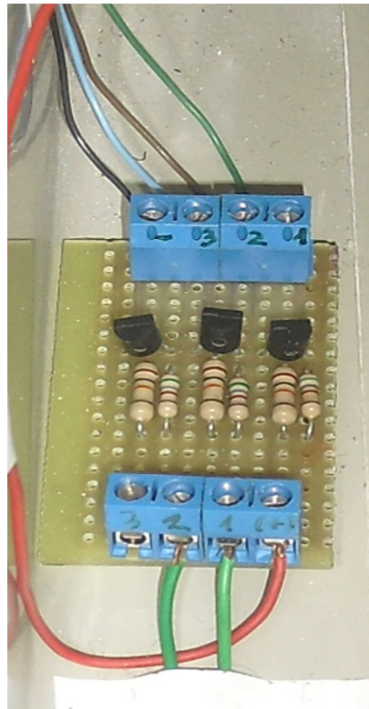


Figura 46: Acondicionamiento de señal de los sensores ópticos

Las entradas y salidas analógicas se han cableado directamente también del cuadro original. Al tratarse de transductores de dos hilos, se ha puesto especial cuidado en la protección de las entradas de tensión y corriente conectando las masas del cuadro y el autómata como muestra la Figura 47:

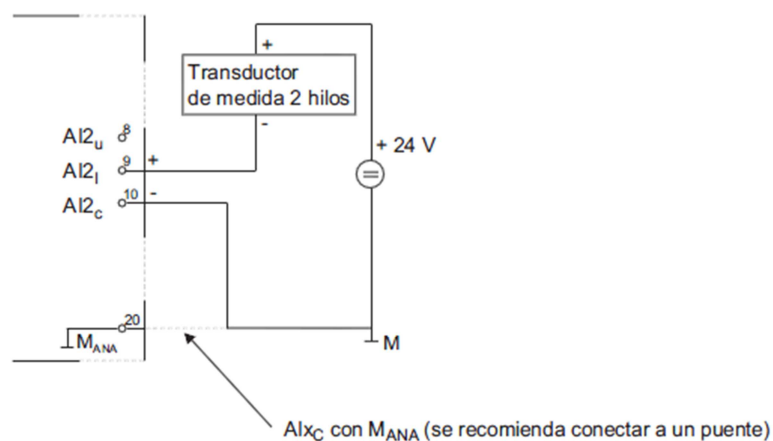


Figura 47: Protección de las entradas analógicas

El esquema de conexión de las entradas, así como todo lo explicado en este punto referido a las Entradas/Señales del autómata, se explicará con más detalle en el próximo capítulo.

### 3.6. El autómata de Siemens.

El autómata utilizado es el *Controlador Programable S7-300* de la casa *Siemens*. El último número indica la serie a la que pertenece el dispositivo, habiendo varias versiones del mismo dentro de dicha serie. Las diferencias entre cada versión vienen dadas por el tipo de *CPU* de la que conste el controlador. Asimismo, cada *CPU* se diferencia de las demás según su velocidad,

memoria, capacidad de comunicación, funciones a realizar (conteo, temporización, comparación...), capacidad de integrar entradas y salidas tanto digitales como analógicas, etc. Para nuestro sistema de control hemos elegido la *CPU 314 C-2 DP*. Además tiene acoplados un módulo de entradas y salidas digitales y un contador. También dispone de una fuente de alimentación de 5 amperios y puede comunicarse mediante los perfiles *MPI* y *DP* por bus de campo. Todos estos elementos están sujetos a un bastidor y también se han comunicado las *E/S* a un entrenador para completar la parte docente, tal y como podemos ver en la imagen siguiente (Figura 48):

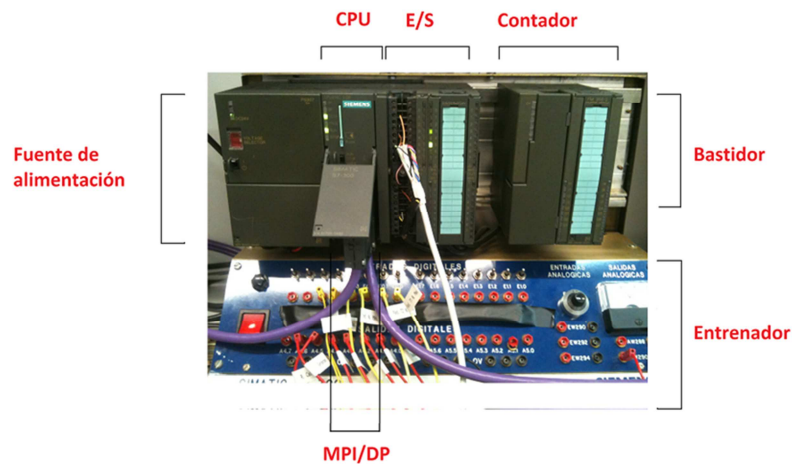


Figura 48: Autómata S7-300; CPU 314 C-2 DP

Como ya comentáramos con anterioridad, el autómata que incorporaba la maqueta, ha sido desconectado para el correcto funcionamiento de nuestro controlador. A continuación veremos sus características técnicas.

#### Características técnicas de la CPU:

- Referencia: 6ES7 314-6CF01-0AB0
- Memoria de trabajo integrada: 48Kbytes
- Memoria de carga: insertable mediante MMC (máx. 8 Mbytes)
- Respaldo: garantizado por la MMC (sin mantenimiento)
- Contador S7: 256
- Contador IEC: Sí
- Temporizadores S7: 256
- Temporizador IEC: Sí
- Marcas: 256 bytes
- Bloques de datos: máx. 511 (de DB 1 a DB 511)
- Áreas de direccionamiento (E/S): máx. 1024 bytes / 1024 bytes

- Imagen de proceso E/S: 128 bytes / 128 bytes
- Canales digitales: máx. 8192
- Canales digitales integrados: 24 DI / 16 DO
- Canales analógicos: máx. 512
- Canales analógicos integrados: 4 + 1 AI / 2 AO
- Comunicación: MPI / PROFIBUS DP
- Lenguaje de programación: KOP / FUP / AWL
- Direcciones entradas digitales integradas: 124.0 a 126.7
- Direcciones salidas digitales integradas: 124.0 a 125.7
- Direcciones entradas analógicas integradas: 752 a 761
- Direcciones salidas analógicas integradas: 752 a 755
- Función PID
- Tensión de alimentación: 24 Vdc.
- Consumo de corriente: 150 mA.
- Potencia disipada: 14 W.

Ahora que ya conocemos las características técnicas más importantes de la *CPU* de nuestro autómat, vamos a conocer el resto de módulos principales: la fuente de alimentación y los módulos de *E/S*. El resto de características técnicas pueden ser consultadas en los anexos.

### 3.6.1. Fuente de alimentación.

La fuente que alimenta al autómat es una fuente de 5 amperios de salida, modelo *PS 307 6ES7307-1EA00-0AA0* (ver *Figura 48*). Puede utilizarse como fuente de alimentación de carga y se recomienda proteger la entrada de línea de red de la fuente con un automático magnetotérmico. La tensión nominal de salida es de 24 Vdc. estabilizada, a prueba de cortocircuitos y marcha en vacío. En la imagen siguiente (*Figura 49*) se muestra el esquema de conexión del dispositivo:

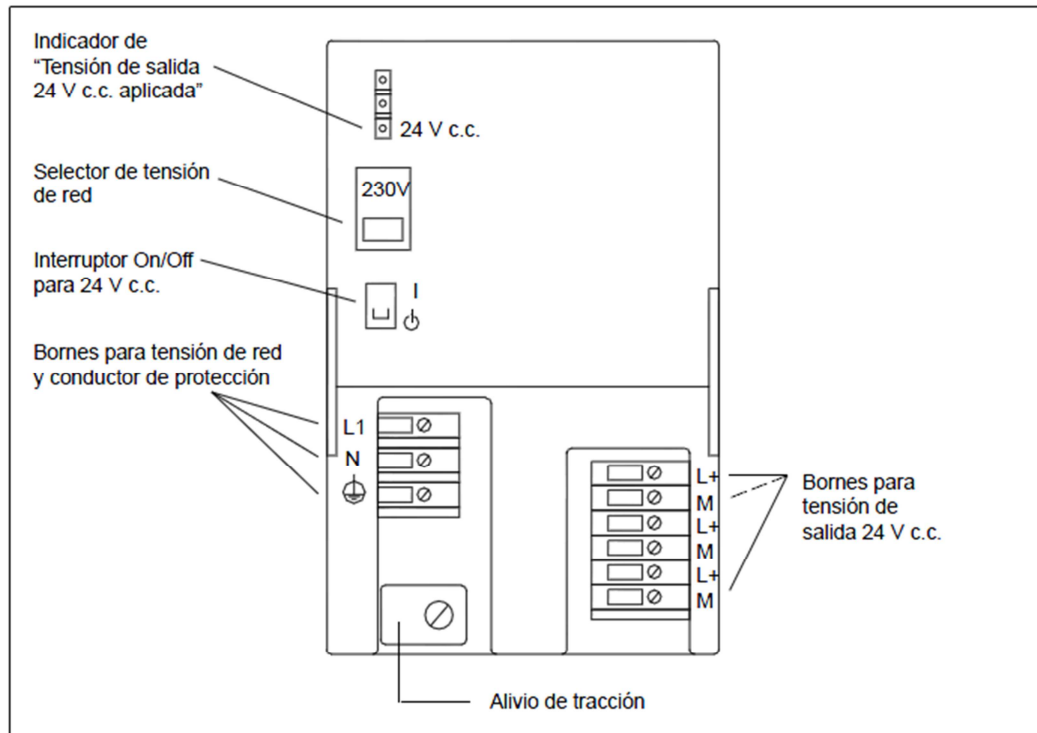


Figura 49: Esquema de conexión de la fuente de alimentación

Mientras que a nivel interno, el aparato está configurado tal y como muestra la *Figura 50*, representando todos los elementos necesarios.

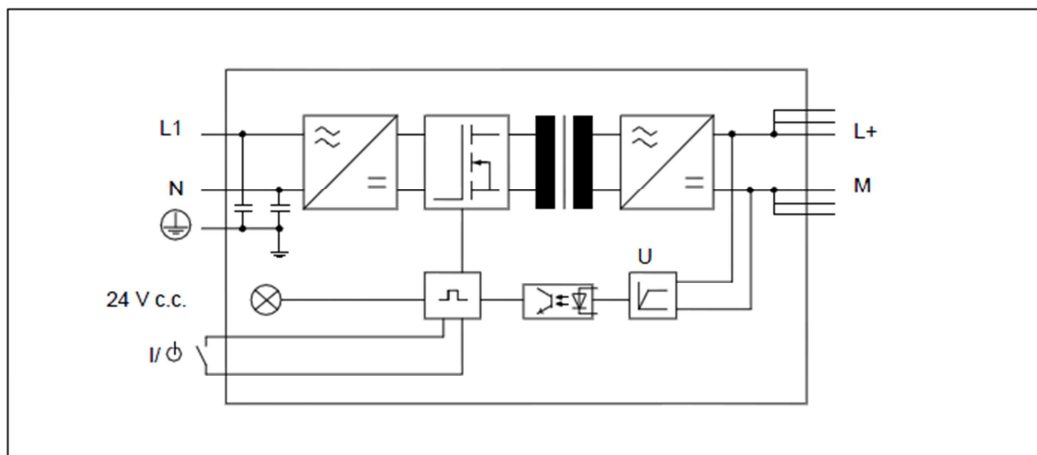


Figura 50: Arquitectura interna de la fuente de alimentación

Por último, como ya hemos hecho anteriormente, vamos a conocer las características técnicas del dispositivo.

#### Características técnicas:

- Tensión de entrada: 120 / 230 Vca.
- Frecuencia de red: 50 o 60 Hz
- Intensidad de entrada: a 120 V – 2 A; a 230 V – 1 A.

- Tensión de salida: 24 Vdc.
- Intensidad de salida: 5 A.
- Rendimiento: 87%
- Potencia absorbida: 138 W.

### 3.6.2. Módulos de E/S.

Como ya hemos visto en las características técnicas de la *CPU*, el autómata consta de 24 DI, 16 DO, 5 AI y 2 AO todas ellas integradas en el dispositivo. En este punto vamos a ver su arquitectura interna y cómo se organizan en el módulo de *E/S* del autómata.

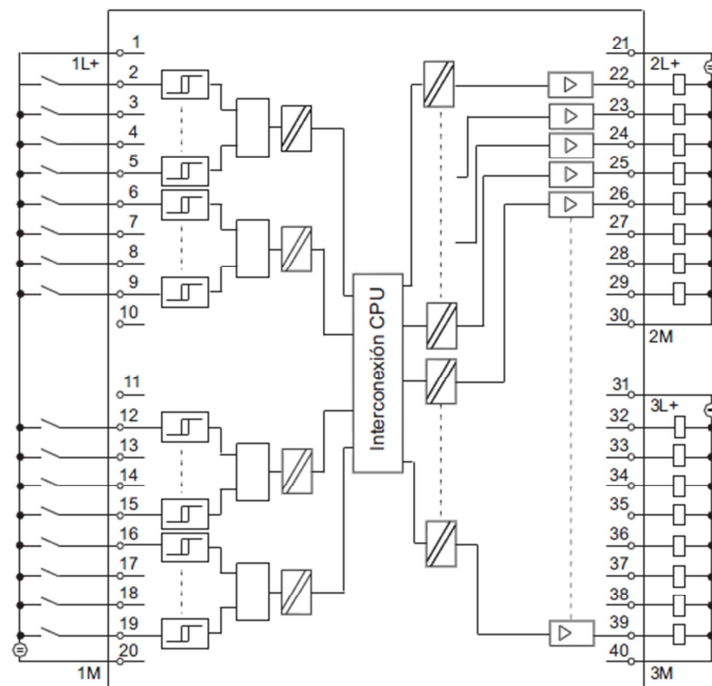
#### Entradas y salidas digitales.

En la *Figura 51* se observa cómo se organizan dentro del módulo de *E/S* y su arquitectura interna integradas en el autómata.

X11 de la CPU 313C-2 PtP/DP  
X12 de la CPU 314C-2 PtP/DP

DI estándar	Entrada alarma	Contador	Posicionamiento <sup>1)</sup>	1 Ø	1L+	2L+	Ø 21	Posicionamiento <sup>1)</sup> digital	analógico	Contad.	DO estándar
X	X	Z0 (A)	A 0	2 Ø	DI+0.0	DO+0.0	Ø 22			V0	X
X	X	Z0 (B)	B 0	3 Ø	DI+0.1	DO+0.1	Ø 23			V1	X
X	X	Z0 (puerta HW)	N 0	4 Ø	DI+0.2	DO+0.2	Ø 24			V2	X
X	X	Z1 (A)	Tast 0	5 Ø	DI+0.3	DO+0.3	Ø 25			V3.1)	X
X	X	Z1(B)	Bero 0	6 Ø	DI+0.4	DO+0.4	Ø 26				X
X	X	Z1 (puerta HW)		7 Ø	DI+0.5	DO+0.5	Ø 27				X
X	X	Z2 (A)		8 Ø	DI+0.6	DO+0.6	Ø 28		CONV EN		X
X	X	Z2 (B)		9 Ø	DI+0.7	DO+0.7	Ø 29		CONV DIR		X
				10 Ø		2M	Ø 30				
				11 Ø		3L+	Ø 31				
X	X	Z2 (puerta HW)		12 Ø	DI+1.0	DO+1.0	Ø 32	R+			X
X	X	Z3 (A)		13 Ø	DI+1.1	DO+1.1	Ø 33	R-			X
X	X	Z3 (B)	1)	14 Ø	DI+1.2	DO+1.2	Ø 34	Rápida			X
X	X	Z3 (puerta HW)		15 Ø	DI+1.3	DO+1.3	Ø 35	Lenta			X
X	X	Z0 (Latch)		16 Ø	DI+1.4	DO+1.4	Ø 36				X
X	X	Z1 (Latch)		17 Ø	DI+1.5	DO+1.5	Ø 37				X
X	X	Z2 (Latch)		18 Ø	DI+1.6	DO+1.6	Ø 38				X
X	X	Z3 (Latch)	1)	19 Ø	DI+1.7	DO+1.7	Ø 39				X
				20 Ø	1M	3M	Ø 40				

**Esquema de principio de la periferia digital integrada de las CPUs 313C/313C-2/314C-2**



*Figura 51: E/S digitales*

Podemos observar en las imágenes anteriores la correspondencia con los bornes de los módulos y las entradas y salidas digitales. Hay que destacar que las entradas digitales están aisladas del núcleo del autómata por medio de optoacopladores. También podemos ver los bornes de tensión y masa (*L+* y *M*) de cada grupo de entradas y salidas. Por último vamos a conocer sus principales características técnicas de estos módulos.

Características técnicas:

- Cantidad de entradas/útiles para funciones tecnológicas: 24/16
- Tensión nominal de carga *L+*: 24 Vdc.
- Separación galvánica entre canales y bus posterior en las entradas.
- Tensión de entrada: 24 Vdc.
- Tensión de entrada para la señal “1”: de 15 a 30 Vdc.
- Tensión de entrada para la señal “0”: de -3 a 5 Vdc.
- Intensidad de entrada en la señal “1”: 9 mA.
- Cantidad de salidas/salidas rápidas: 16/4
- Intensidad de salida en las DO para señal “1”: 0.5 A.
- Intensidad de salida en las DO para señal “0”: 0.5 mA.
- Resistencia de carga en las DO: de 48 a 4 Kohmios.



## Entradas y salidas analógicas.

En la *Figura 52* se observa cómo se organizan dentro del módulo de E/S y su arquitectura interna integradas en el autómata.

Estándar	Posicionar	X11				DI estándar	Entrada alarma
		1 Ø			Ø 21		
AI (Ch0)	V	2 Ø	PEWx+0	DI+2.0	Ø 22	X	X
	I	3 Ø		DI+2.1	Ø 23	X	X
	C	4 Ø		DI+2.2	Ø 24	X	X
AI (Ch1)	V	5 Ø	PEWx+2	DI+2.3	Ø 25	X	X
	I	6 Ø		DI+2.4	Ø 26	X	X
	C	7 Ø		DI+2.5	Ø 27	X	X
AI (Ch2)	V	8 Ø	PEWx+4	DI+2.6	Ø 28	X	X
	I	9 Ø		DI+2.7	Ø 29	X	X
	C	10 Ø		4M	Ø 30		
AI (Ch3)	V	11 Ø	PEWx+6		Ø 31		
	I	12 Ø			Ø 32		
	C	13 Ø			Ø 33		
PT 100 (Ch4)		14 Ø	PEWx+8		Ø 34		
		15 Ø			Ø 35		
AO (Ch0)	V	16 Ø	PAW x+0		Ø 36		
	A	17 Ø			Ø 37		
AO (Ch1)	V	18 Ø	PAW x+2		Ø 38		
	A	19 Ø			Ø 39		
		20 Ø	M <sub>aux</sub>		Ø 40		

1) sólo CPU 314C-2

Esquema de principio de la periferia analógica/digital integrada de las CPUs 313C/314C2

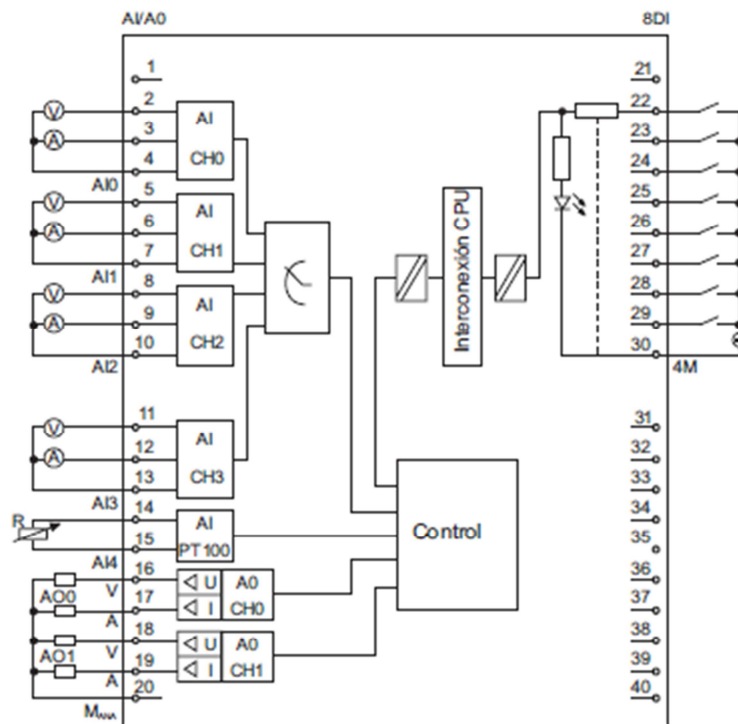


Figura 52: E/S analógicas

Como se puede ver en la imagen, cada canal de entrada analógico, dispone de entradas tanto por tensión como por corriente. Por su parte, los canales de salida también disponen de salidas tanto por tensión como por corriente.

Otro aspecto importante es la protección de las entradas analógicas cuando estén conectadas a cualquier transductor. Como ya viéramos anteriormente, cuando queramos

proteger las entradas, ya sea con transmisores de 2 o 4 hilos, deberemos seguir el esquema mostrado en la siguiente imagen (Figura 53):

#### Protección de las entradas de tensión e intensidad

Las siguientes figuras muestran la protección de las entradas de tensión e intensidad con un transductor de medida a 2/4 hilos.

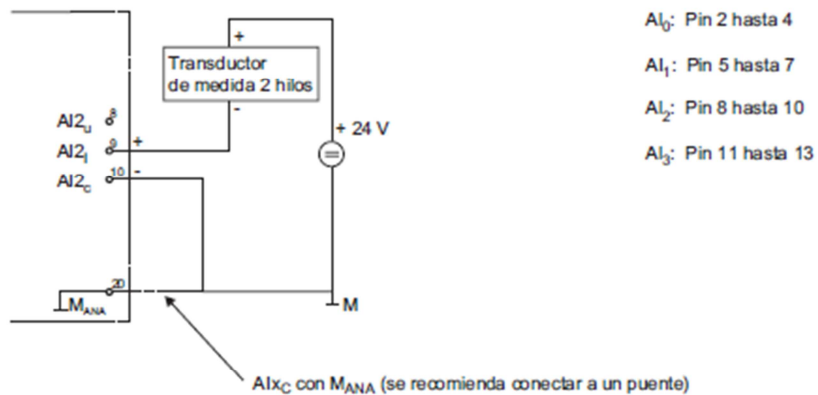


Figura 6-1 Protección de una entrada analógica de tensión e intensidad en la CPU 313C/314C-2 con un transductor de medida a 2 hilos

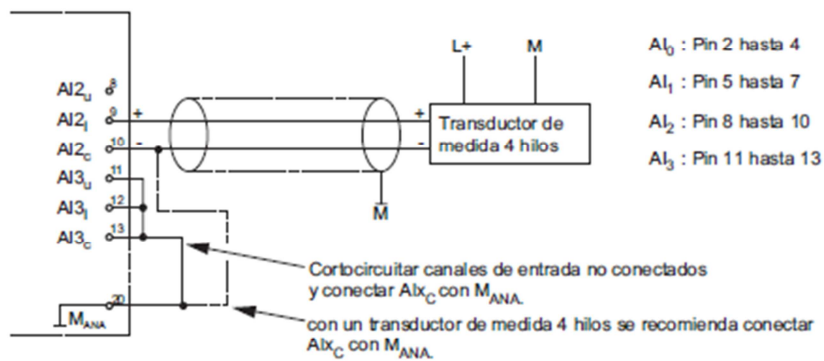


Figura 6-2 Protección de una entrada analógica de tensión e intensidad en la CPU 313C/314C-2 con un transductor de medida a 4 hilos

Figura 53: Protección de las entradas analógicas con transductores de 2 y 4 hilos

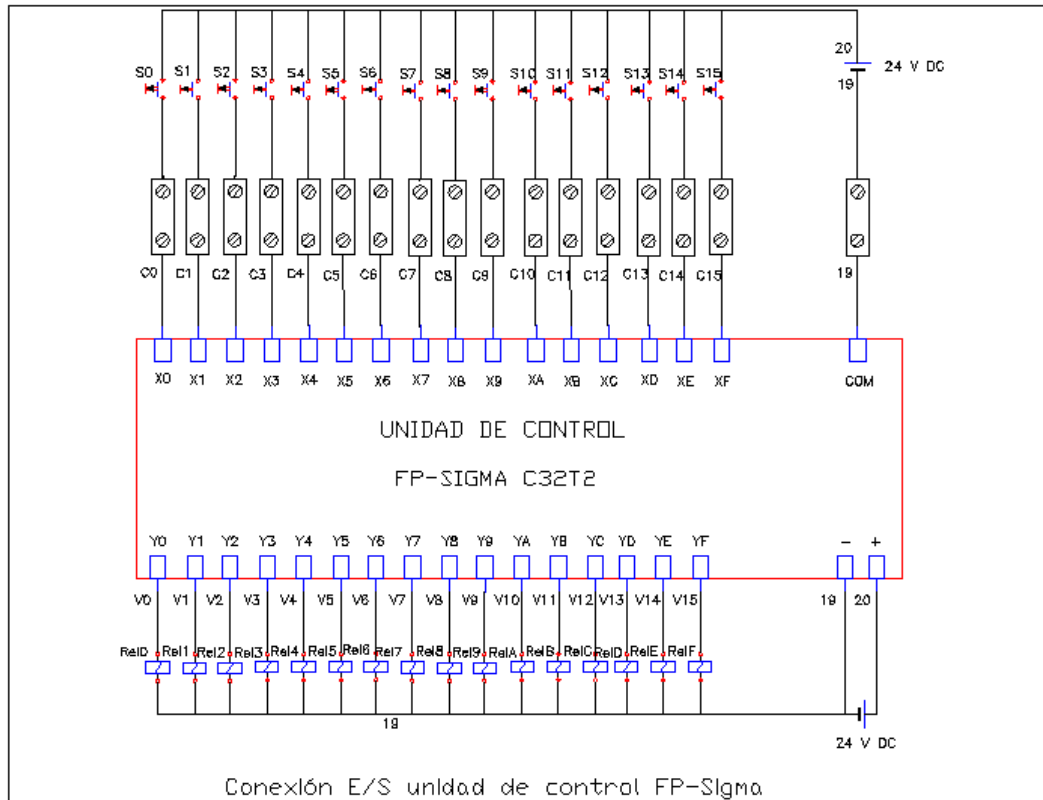
#### Características técnicas:

- Cantidad de entradas: 4 canales de entrada por V e I; 1 canal de entrada de resistencia.
- Separación galvánica entre canales y bus posterior.
- Principio de medida: codificación momentánea.
- Resolución: 11 bits más signo.
- Tiempo de ejecución: 1 ms.
- Exactitud de repetición: +/- 0.06 %
- Tensión de entrada: +/- 10 V o de 0 a 10 V
- Intensidad de entrada: 4 – 20 mA; +/- 20 mA; de 0 a 20 mA.
- Termorresistencia: Pt100

- Cantidad de salidas: 2
- Tensión nominal de carga de las salidas: 24 Vdc.

### 3.7. Duplicado Entradas/Salidas del cuadro original.

Tomando el esquema de conexión de las E/S del cuadro original, vamos a mostrar cómo hemos hecho el duplicado de las mismas, necesario para nuestro proyecto. En la *Figura 54* se muestra el esquema del cuadro original:



*Figura 54: Esquema E/S del cuadro original*

Se ha utilizado como fuente de alimentación auxiliar el modulo FP PSA2 con un voltaje nominal de 24 Vdc. Para las entradas se han cableado los cuatro sensores de nivel de los depósitos superiores según la tabla (*Figura 55*):

Tipo	Valor	Ref.	Denominación
Entrada Digital	X0	S0	Sensor nivel máx. Tanque 1 frío
Entrada Digital	X1	S1	Sensor nivel mín. Tanque 1 frío
Entrada Digital	X2	S2	Sensor nivel máx. Tanque 2 calor
Entrada Digital	X3	S3	Sensor nivel mín. Tanque 2 calor

*Figura 55: Referencia de las entradas*

Para las salidas se han utilizado relés (desde Rel. 0 a Rel. F). Los Relés utilizados son marca *PHOENIX CONTAC* cuya tensión nominal de la bobina es de 24 Vdc. En el siguiente plano (Figura 56) se muestra el contacto de relés.

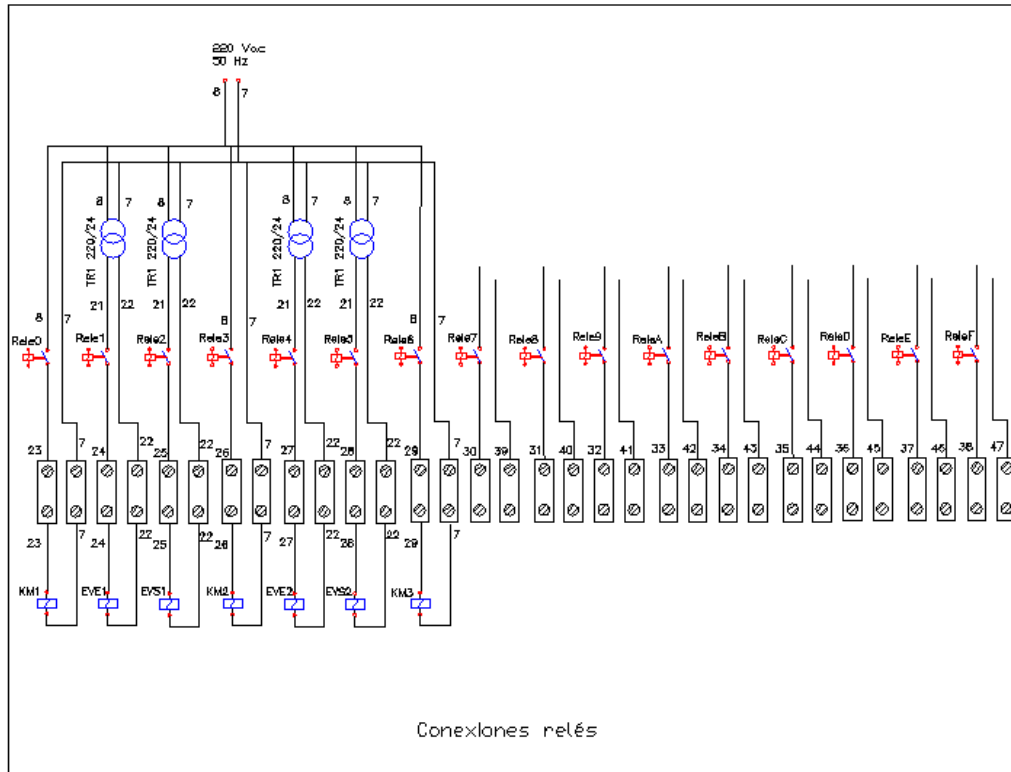


Figura 56: Conexión de relés

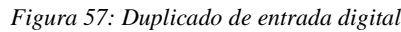
Los contactos de los relés 0, 3, 6 cierran el circuito alimentando las bobinas de los contactores de tensión nominal 220 Vac. Los contactos de los relés 1, 2, 4, 5 cierran el circuito alimentando a las bobinas de las electroválvulas cuya tensión nominal es de 24 Vac. Dicha tensión la obtenemos de un transformador monofásico 220Vac/24Vac instalado para tal fin.

Después de conocer el esquema del cuadro original de las entradas y salidas, vamos a tomar como ejemplo el duplicado de una entrada y salida digital y analógica. Para ello nos vamos a servir de los planos anteriores para configurar un nuevo esquemático con la *E/S* duplicada.

### 3.7.1. Duplicado E/S.

#### Entrada digital.

Tomaremos como ejemplo la entrada digital *S0 Sensor nivel máx. Tanque 1 frío* (ver Figura 55); nos referimos al flotador-interruptor que indica el nivel máximo del tanque 1. En el borne de la regleta donde está conectado (ver Figura 54), duplicaremos la salida del sensor conectando otro cable y llevándolo a nuestro autómatas, tal y como puede verse en la Figura 57:



Por otro lado, debemos tener en cuenta la siguiente consideración. Al tomar la alimentación de 24 voltios del cuadro original, bajo ningún concepto debemos conectar la alimentación de la fuente del autómatas de *Siemens* si no queremos conectar las fuentes de alimentación en paralelo, pudiendo llegar a quemarlas. No obstante, si conectaremos las masas del cuadro y el PLC de *Siemens* para tener la misma masa en todo el sistema y cerrar el circuito.

Las salidas digitales en el *PLC* del cuadro están cableadas a una serie de relés que se activarán cuando el autómata *FP-Sigma* emita tensión en la salida. Por otro lado, para las salidas de nuestro controlador, también se ha optado por incluir relés para su activación, aislando así las alimentaciones del cuadro original y de nuestro autómata como muestra la *Figura 58*:

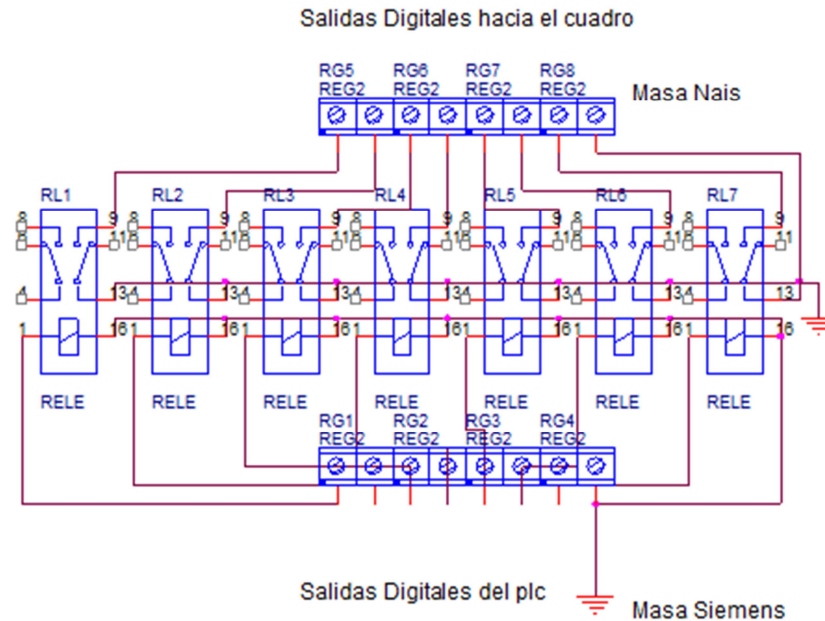


Figura 58: Relés para aislar las masas del cuadro y del PLC de Siemens

De esta forma, para duplicar la salida, hemos llevado las entradas de tensión a los relés del cuadro a nuestra regleta, y de ahí las hemos conectado a los interruptores de nuestros relés. La masa del cuadro original también se ha conectado a los otros bornes de los interruptores de los relés. Las bobinas se han conectado a las salidas digitales de nuestro *PLC* y a su masa para cerrar el circuito, como puede verse en la imagen a continuación (Figura 59):

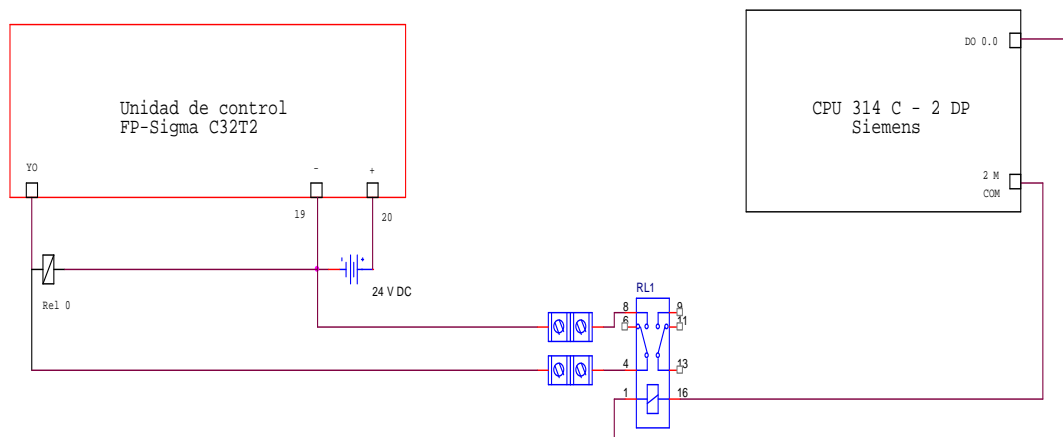


Figura 59: Duplicado de salida digital

### Entrada analógica.

Para explicar el duplicado de las entradas analógicas vamos a tomar como ejemplo la sonda de nivel (ver Figura 41). La entrada analógica puede ser por tensión o por corriente. Nosotros hemos elegido esta última opción. Hemos cableado la entrada por corriente de la sonda de nivel en el cuadro original llevándola a nuestro autómat. También hemos cableado la masa del cuadro y la hemos conectado con la masa de nuestro *PLC* como medida de

protección, ya que la sonda se trata de un transductor a dos hilos. La *Figura 60* muestra el duplicado.

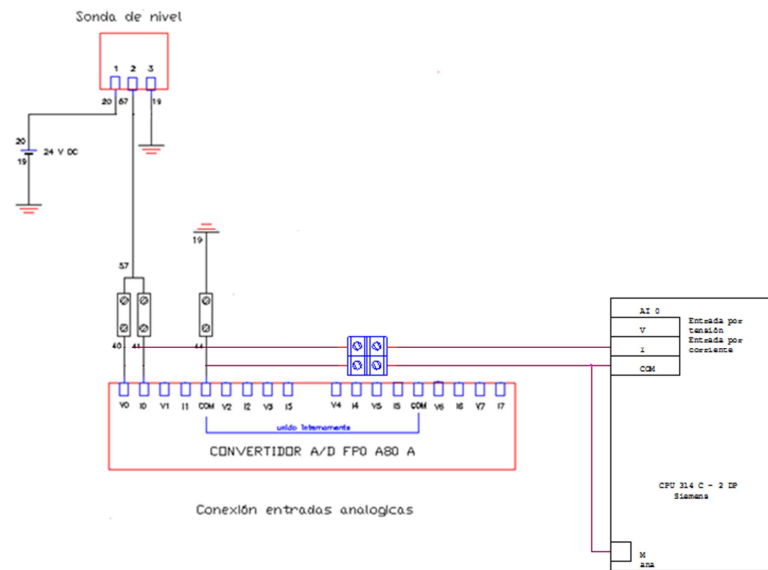


Figura 60: Duplicado de entrada analógica

### 3.8. Sensor inteligente SITRANS Probe LU.

Para medir el nivel del depósito de mezcla, además del sensor analógico anteriormente explicado, se ha utilizado el sensor inteligente de *Siemens*, *SITRANS Probe LU*. Este sensor funciona mediante la tecnología de ultrasonidos para obtener la medición. Los resultados se pueden observar mediante un display incorporado en el sensor o a través del software *SIMATIC PDM* en la estación de control. En la *Figura 61* podemos ver el sensor colocado en la maqueta.



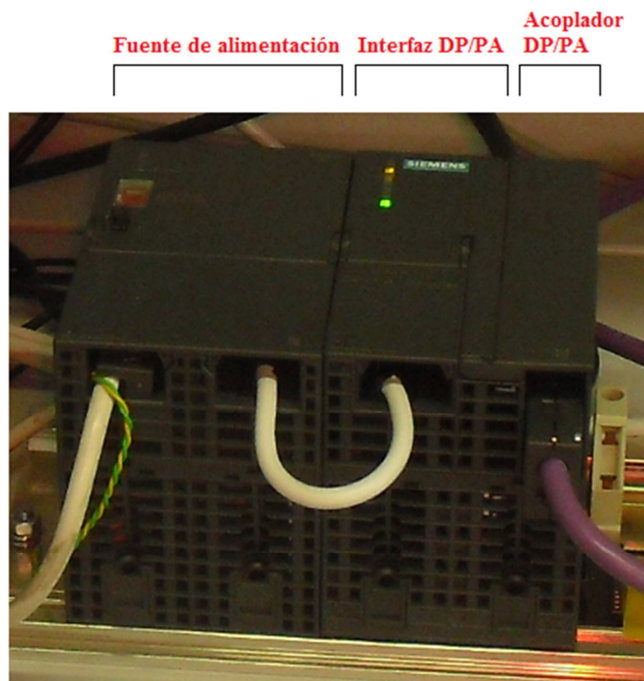
Figura 61: Detalle del sensor en la maqueta

Las características de este dispositivo serán explicadas en el capítulo quinto de este proyecto. Asimismo se podrán consultar los anexos para más información del sensor.

### 3.8.1. Pasarela PROFIBUS DP/PA.

Para poder comunicar el autómatas con el sensor y recabar información de este último, es necesario incorporar a nuestro sistema de control una pasarela *DP/PA*. El sensor inteligente está diseñado para trabajar en entornos hostiles, es por ello que utiliza el estándar de comunicación *PROFIBUS PA*. Mientras que por su parte, el autómatas se comunica mediante *PROFIBUS DP*. Por ello, se hace necesario el uso de esta pasarela.

El dispositivo consta de dos partes visiblemente diferenciadas. El módulo de interfaz *DP/PA* y el acoplador de los dos estándares. Además lleva acoplada una fuente de alimentación de 5 amperios como podemos ver en la imagen siguiente (*Figura 62*):



*Figura 62: Pasarela PROFIBUS DP/PA*

Por su parte, para comunicar el autómatas con el sensor, necesitamos conectar los dos estándares en el acoplador. En la *Figura 63* podemos ver cómo están conectados los dos perfiles, comunicando autómatas con sensor.





Figura 63: Acoplador DP/PA

De esta forma, tras las modificaciones realizadas en la maqueta, nuestro sistema quedaría tal y como vemos en la *Figura 64*:

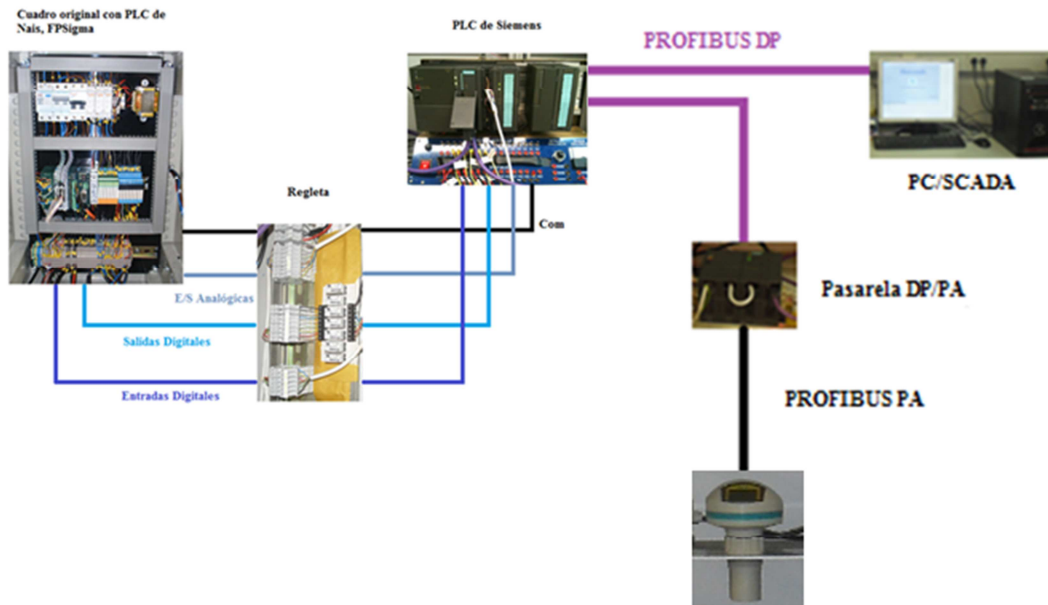


Figura 64: Esquema del sistema

# Capítulo 4: Conexiones y Software.

En este capítulo vamos a conocer las conexiones realizadas con el autómata, sus necesidades y el software requerido para el funcionamiento del sistema.

## 4.1. Introducción.

Como ya explicáramos en el capítulo anterior, el autómata que hemos utilizado en nuestro proyecto es el PLC de *Siemens 314 C-2 DP*. Este dispositivo dispone de un módulo de *E/S* integradas, siendo el tipo y la cantidad de éstas tal y como se describe a continuación:

- Entradas digitales útiles para funciones: 16
- Salidas digitales: 16
- Entradas analógicas: 4 + 1
- Salidas analógicas: 2

Por otro lado, si observamos la *Figura 1* atentamente, podremos constatar los elementos que intervienen en el sistema y la cantidad de entradas y salidas necesarias para regularlo.

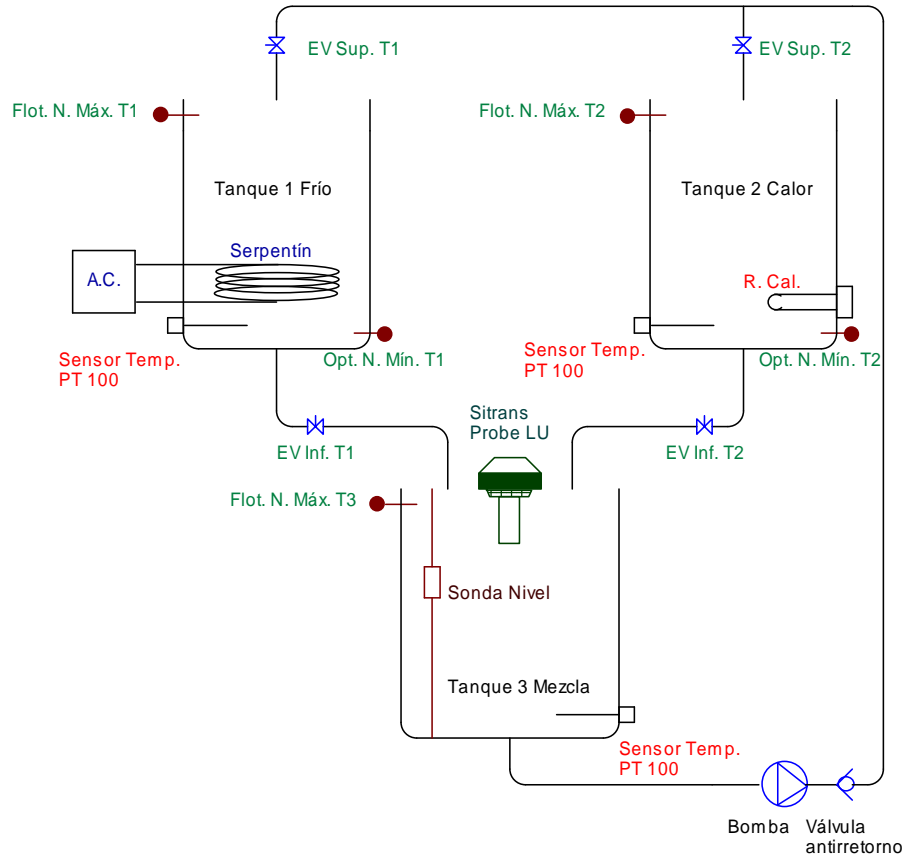


Figura 1: Esquema del sistema

Es sencillo pues, deducir qué elementos serán entradas y cuáles salidas. Si recordamos el apartado de sensorización del capítulo anterior, podremos saber también de qué tipo, si digitales o analógicas. Por lo tanto, todos aquellos dispositivos encargados de recabar información del estado de la maqueta, como los sensores, serán considerados como entradas mientras que por el contrario, todos aquellos aparatos que actúen directamente sobre el sistema, serán considerados como salidas del autómata. De esta forma, podemos clasificarlos según la tabla (Figura 2):

	Digital	Analógica
<b>Entrada</b>	Flotadores y sensores ópticos	Sonda de nivel y Pt100
<b>Salida</b>	Bomba, electroválvulas, serpentín y resistencia calorífica	

Figura 2: Clasificación de los dispositivos del sistema como E/S del autómata

Hecho esto, la disposición de las entradas y salidas junto con los aparatos que intervienen en la maqueta, quedará de la forma:

- Entradas digitales (5):
 

Flot. N. Máx. T1;	(flotador-interruptor)
Flot. N. Máx. T2;	(flotador-interruptor)
Flot. N. Máx. T3;	(flotador-interruptor)
Opt. N. Mín. T1;	(sensor óptico)
Opt. N. Mín. T2;	(sensor óptico)
  
- Salidas digitales (7):
 

Bomba;	(motor)
EV. Sup. T1;	(electroválvula)
EV. Sup. T2	(electroválvula)
EV. Inf. T1;	(electroválvula)
EV. Inf. T2;	(electroválvula)
R. Cal.;	(resistencia)
Serpentín;	(máquina de frío)
  
- Entradas analógicas (4):
 

Sonda de nivel;	(sensor de nivel)
Pt 100 T1;	(sensor de temperatura)
Pt 100 T2;	(sensor de temperatura)
Pt 100 T3;	(sensor de temperatura)

Mención aparte recibirá el sensor inteligente por ultrasonidos *SITRANS Probe LU*. Este dispositivo será explicado en el capítulo siguiente. No obstante, se explicará en este capítulo su estándar de comunicación *PROFIBUS PA\**.

Una vez que sabemos el número de *E/S* que vamos a necesitar y cómo se clasifican según los dispositivos del sistema a los que hacen referencia, la estructura hardware del conexionado con el autómat quedará de la forma en la que se muestra en la *Figura 3*:

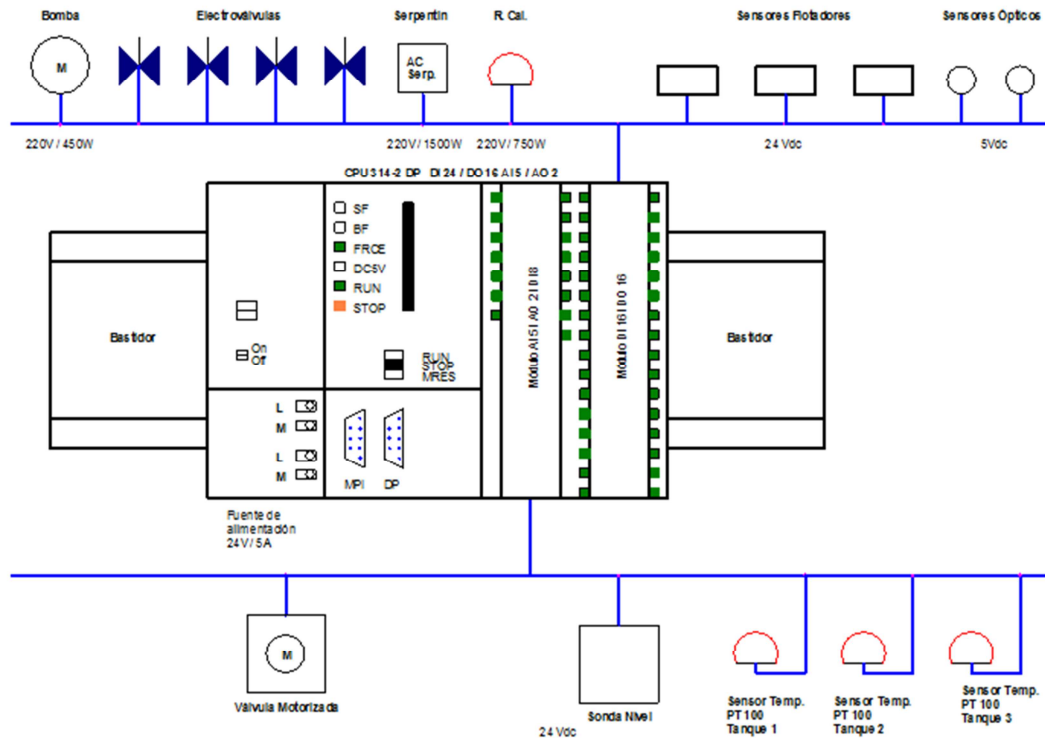


Figura 3: Arquitectura hardware de las E/S del sistema

## 4.2. Asociación de Entradas y Salidas del sistema.

Una vez que sabemos las entradas y salidas necesarias y qué elementos van a intervenir en el sistema, el siguiente paso será asociar dichos elementos con una entrada o salida del autómat, según corresponda. Es importante no confundir esta asociación, con la asociación de variables que haremos a la hora de programar y configurar la interfaz de control del sistema.

### 4.2.1. Entradas y salidas digitales.

Si observamos la *Figura 4*, veremos el módulo incorporado de entradas y salidas digitales del autómat.

X11 de la CPU 313C-2 PtP/DP  
X12 de la CPU 314C-2 PtP/DP

DI estándar	Entrada alarma	Contador	Posicionamiento <sup>1)</sup>	1 0	1L+	2L+	0 21	Posicionamiento <sup>1)</sup>	Contad.	DO estándar
X	X	Z0 (A)	A 0	2 0	DI+0.0	DO+0.0	0 22		V0	X
X	X	Z0 (B)	B 0	3 0	DI+0.1	DO+0.1	0 23		V1	X
X	X	Z0 (puerta HW)	N 0	4 0	DI+0.2	DO+0.2	0 24		V2	X
X	X	Z1 (A)	Tast 0	5 0	DI+0.3	DO+0.3	0 25		V3 1)	X
X	X	Z1 (B)	Bero 0	6 0	DI+0.4	DO+0.4	0 26			X
X	X	Z1 (puerta HW)		7 0	DI+0.5	DO+0.5	0 27			X
X	X	Z2 (A)		8 0	DI+0.6	DO+0.6	0 28	CONV EN		X
X	X	Z2 (B)		9 0	DI+0.7	DO+0.7	0 29	CONV DIR		X
				10 0		2M	0 30			
				11 0		3L+	0 31			
X	X	Z2 (puerta HW)		12 0	DI+1.0	DO+1.0	0 32	R+		X
X	X	Z3 (A)		13 0	DI+1.1	DO+1.1	0 33	R-		X
X	X	Z3 (B)	>1)	14 0	DI+1.2	DO+1.2	0 34	Rápida		X
X	X	Z3 (puerta HW)		15 0	DI+1.3	DO+1.3	0 35	Lenta		X
X	X	Z0 (Latch)		16 0	DI+1.4	DO+1.4	0 36			X
X	X	Z1 (Latch)		17 0	DI+1.5	DO+1.5	0 37			X
X	X	Z2 (Latch)		18 0	DI+1.6	DO+1.6	0 38			X
X	X	Z3 (Latch)	1)	19 0	DI+1.7	DO+1.7	0 39			X
				20 0	1M	3M	0 40			

Figura 4: Módulo de E/S digitales

Vemos claramente que las entradas y las salidas se agrupan por bytes de programación. Es decir, cada grupo de entradas y salidas está compuesto por ocho unidades (bits). En total en el módulo tenemos dos grupos de entradas y dos grupos de salidas.

Por otro lado, también observamos a qué borne físico del módulo corresponde cada entrada. En las entradas digitales, la alimentación (1L+) y los comunes de cada una (1M) están conectados internamente. Para las salidas, se han conectado la alimentación y el común internamente en dos grupos de ocho (2L+ y 2M; 3L+ y 3M) tal y como muestra la *Figura 5*. Recordemos que la alimentación viene dada por los bornes de tensión de salida de la fuente PS 307 de 5 amperios que ya vimos en el capítulo anterior.

#### Esquema de principio de la periferia digital integrada de las CPUs 313C/313C-2/314C-2

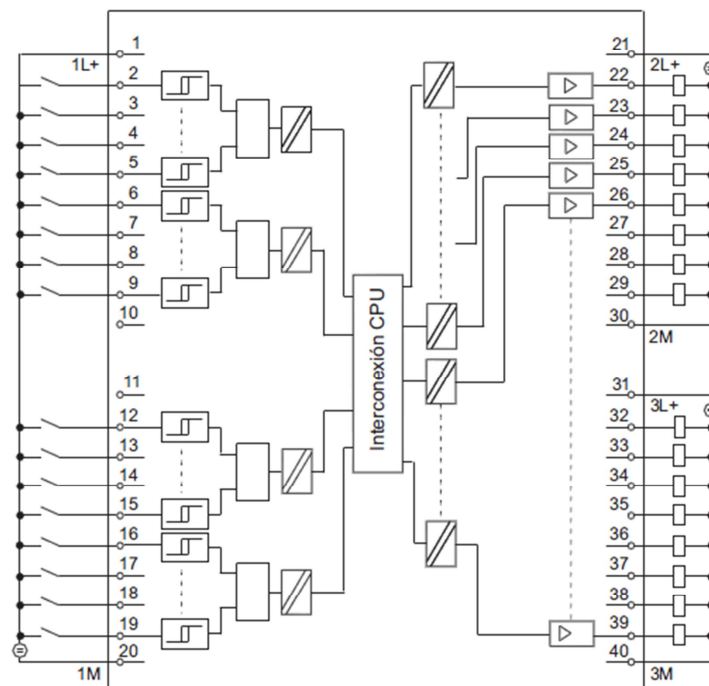


Figura 5: Conexión interno de las E/S digitales

Conocido como se distribuye físicamente el módulo de *E/S*, el siguiente paso será asociar cada borne de entrada al dispositivo adecuado. En la tabla a continuación (*Figura 6*) se muestra la correspondencia de cada entrada y salida con su dispositivo en el sistema físico.

Dispositivo	Borne	Entrada	Salida
Flot. N. Máx. T1	2	DI 0.0	/
Flot. N. Máx. T2	3	DI 0.1	/
Opt. N. Mín. T1	4	DI 0.2	/
Opt. N. Mín. T2	5	DI 0.3	/
Flot. N. Máx. T3	6	DI 0.4	/
Bomba	22	/	DO 0.0
EV. Sup. T1	23	/	DO 0.1
EV. Inf. T1	24	/	DO 0.2
EV. Sup. T2	25	/	DO 0.3
EV. Inf. T2	26	/	DO 0.4
R. Cal.	27	/	DO 0.5
Serpentín	28	/	DO 0.6

Figura 6: Asociación de E/S digitales

#### 4.2.2. Entradas y salidas analógicas.

En la *Figura 7*, veremos el módulo incorporado de entradas y salidas analógicas del autómat.

		X11									
Estándar	Posicionar	1	2	3	4	5	6	7	8	9	10
AI (Ch0)	V	1	2	3	4	5	6	7	8	9	10
	I	11	12	13	14	15	16	17	18	19	20
	C	21	22	23	24	25	26	27	28	29	30
AI (Ch1)	V	1	2	3	4	5	6	7	8	9	10
	I	11	12	13	14	15	16	17	18	19	20
	C	21	22	23	24	25	26	27	28	29	30
AI (Ch2)	V	1	2	3	4	5	6	7	8	9	10
	I	11	12	13	14	15	16	17	18	19	20
	C	21	22	23	24	25	26	27	28	29	30
AI (Ch3)	V	1	2	3	4	5	6	7	8	9	10
	I	11	12	13	14	15	16	17	18	19	20
	C	21	22	23	24	25	26	27	28	29	30
PT 100 (Ch4)		15	16	17	18	19	20	21	22	23	24
AO (Ch0)	valor manip. 0	16	17	18	19	20	21	22	23	24	25
AO (Ch1)	V	16	17	18	19	20	21	22	23	24	25
	A	16	17	18	19	20	21	22	23	24	25

Esquema de principio de la periferia analógica/digital integrada de las CPUs 313C/314C2

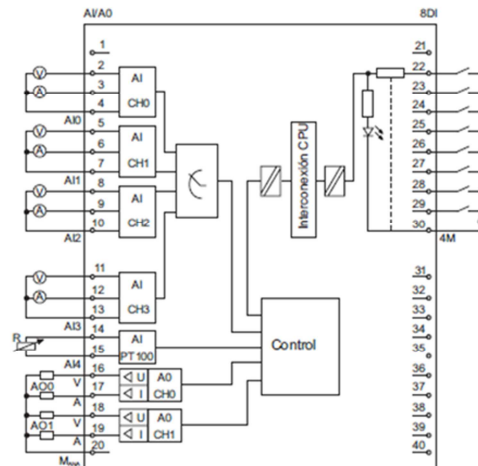


Figura 7: Módulo de E/S analógicas

A diferencia de las entradas y salidas digitales, las analógicas se agrupan por canales. Al encontrarse conectadas con transductores de dos o cuatro hilos, tienen la capacidad de establecer su entrada o salida por tensión o intensidad. Como vemos en la figura anterior, el común de cada canal se encuentra cableado internamente a la masa del sistema como medida de protección (recordemos el capítulo anterior, protección de entradas analógicas). La asociación de entradas y salidas se muestra en la tabla siguiente (*Figura 8*):

Dispositivo	Borne (I)	Entrada	Salida
Sonda de nivel	3	PEW 0	/
Pt100 T1	6	PEW 2	/
Pt100 T2	9	PEW 4	/
Pt100 T3	12	PEW 6	/

*Figura 8: Asociación de E/S analógicas*

### 4.3. Conexionado de Entradas y Salidas.

Una vez que ya sabemos cuántas *E/S* vamos a utilizar, su nomenclatura y a qué dispositivo están asociadas, vamos a mostrar gráficamente su conexionado con el autómata.

#### Entradas/Salidas digitales.

En la *Figura 9* se detalla el esquema de conexión de las entradas y salidas digitales del sistema. Nótese que los dos sensores ópticos (entradas 4 y 5) están alimentados por una fuente de 5 voltios. Como ya vimos en el capítulo anterior, era necesario adecuar la tensión de salida de los sensores ópticos. Esto lo lográbamos con una pequeña placa compuesta por un transistor y dos resistencias por sensor óptico (*Figura 10*).

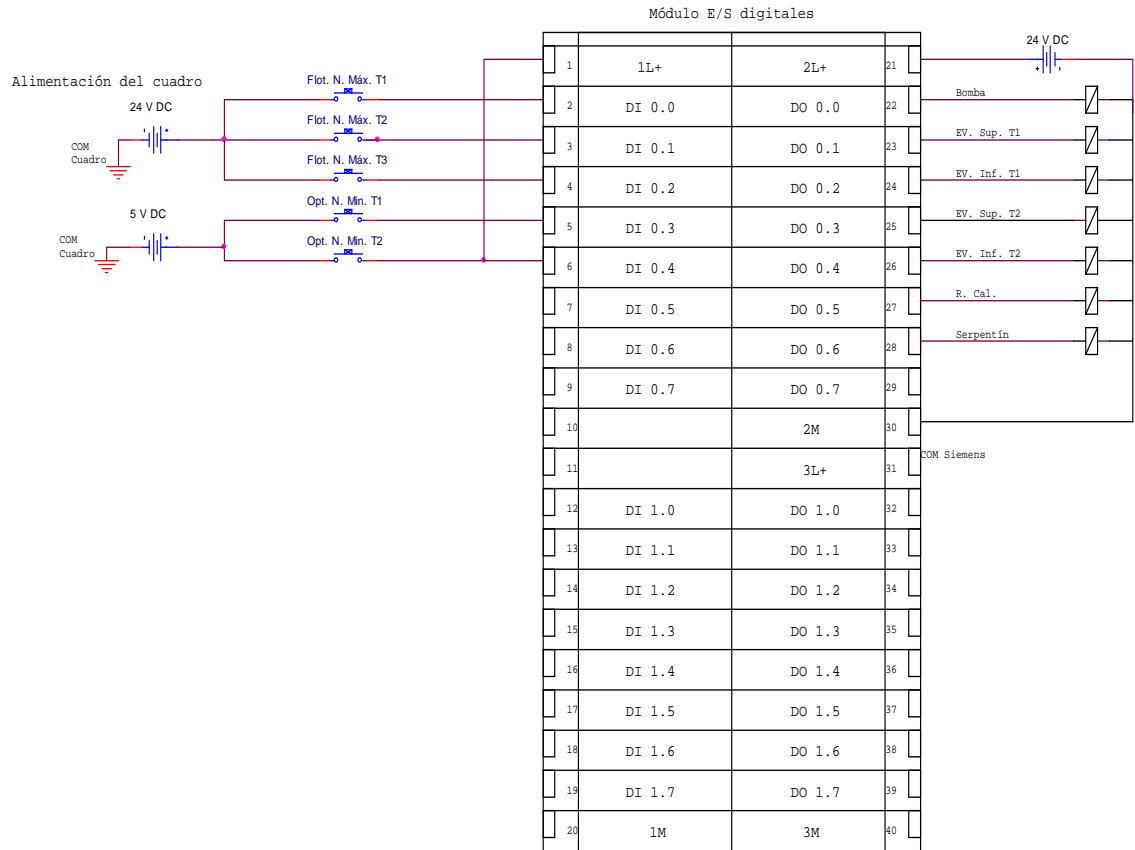


Figura 9: Conexionado de las E/S digitales

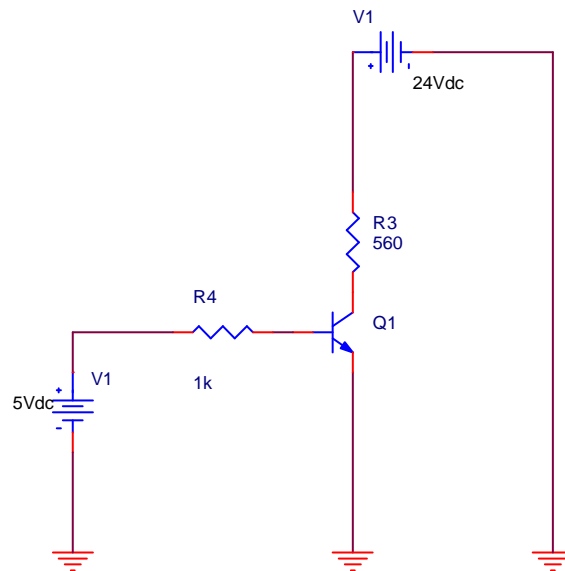


Figura 10: Acondicionamiento señal de ópticos

El transistor  $Q1$  trabajará tanto en corte como en saturación. Mientras la corriente de base  $I_b$  sea muy pequeña, el transistor se encontrará en corte y la tensión  $V_{ce}$  será muy grande. El transistor entrará en saturación cuando se le aplique una intensidad mayor a su base, por lo que la tensión  $V_{ce}$  será muy pequeña. De esta forma conseguimos obtener una tensión de salida del óptico lo suficientemente grande como para activar la entrada del autómeta.



Entradas analógicas.

A continuación, vamos a ver cómo conexionar las entradas analógicas. En la *Figura 11* se muestra de forma esquemática todo el conexionado.

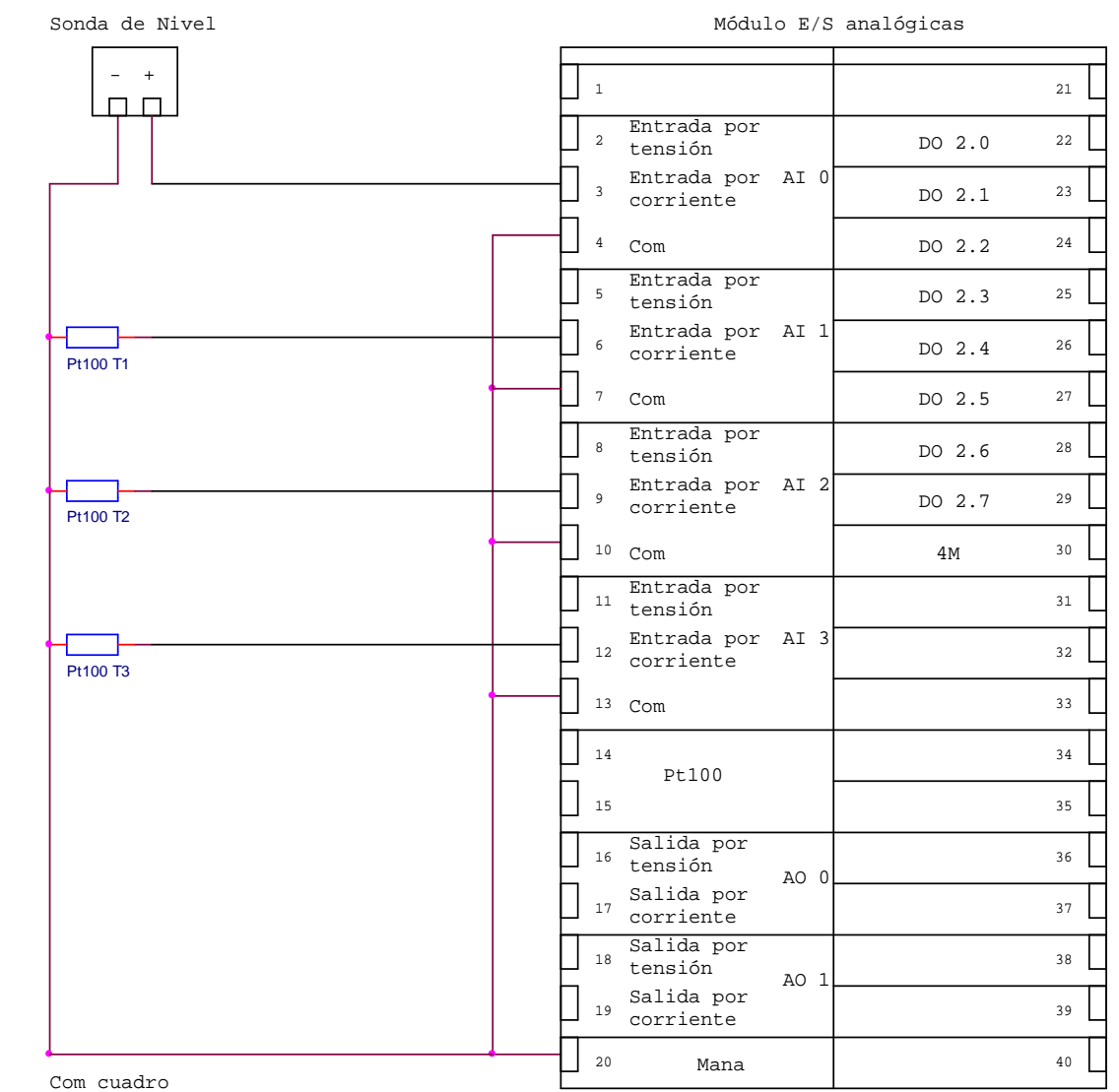


Figura 11: Conexionado entradas analógicas

Hay que recordar una vez más, que el común del cuadro y del *PLC* de *Siemens* se conectan para proteger las entradas.

## 4.4. Conexión y comunicación con SITRANS Probe LU.

Este dispositivo tiene la capacidad de comunicarse mediante dos estándares o protocolos: *HART* o *PROFIBUS*. En este apartado nos vamos a centrar en la comunicación digital con el estándar *PROFIBUS* que es el que hemos utilizado en nuestro sistema. Además, para poder configurar el aparato, necesitamos la tarjeta de comunicación *CP 5611* (ver anexos).

Dado que el sensor está diseñado para trabajar en entornos hostiles (temperaturas altas, por ejemplo), éste sólo se comunica mediante el perfil *PROFIBUS PA*, adaptado a estas circunstancias. Esto a priori puede parecer un problema, ya que nuestro autómatas sólo puede comunicarse a través de *MPI* o *PROFIBUS DP*. Descartado el perfil *MPI* por razones obvias, para solventar esta dificultad, se hace necesario por tanto, la inclusión de una pasarela *DP/PA*; tal y como hemos visto en el anterior capítulo.

### 4.4.1. PROFIBUS.

*PROFIBUS*, derivado de las palabras *Process Field Bus* es un estándar de comunicación para buses de campo. Un bus de campo, a groso modo, es un sistema de intercambio de información capaz de simplificar en gran medida la instalación de aparatos y dispositivos. Este bus se basa en la comunicación controlada entre maestro-esclavo. Definimos de manera particular estos dispositivos como:

- Dispositivos maestros (*Master Devices*). Entre estas estaciones activas rota un permiso de acceso y control que les permite enviar mensajes sin necesidad de petición.
- Dispositivos secundarios (*Slaves Devices*). Periféricos asignados a los maestros. Consisten en una serie de dispositivos lo suficientemente inteligentes como para seguir las normas del protocolo, entre los que podemos encontrar: sensores, actuadores tipo relé, convertidores de frecuencia, electroválvulas, etc. Su papel es pasivo, pudiendo sólo transmitir cuando se les ha realizado una petición previa. Suelen ocupar poco tiempo de comunicación pero son muy numerosos.
- Los datos que se desplazan por el canal físico son de 5 tipos diferentes:
  - Datos de entrada y de salida al proceso.
  - Funciones de diagnóstico y verificación.
  - Configuración de dispositivos.
  - Programas entre los controladores.
  - Parámetros de control.

Además de la línea de comunicación al dispositivo se le debe de dotar de corriente eléctrica que se lleva de manera conjunta con esta línea de datos.

Las principales características de *PROFIBUS* son, según describe la norma *EN 50170 (DIN 19245)* son:

- Abierto. *PROFIBUS* no pertenece a ninguna compañía, está controlado por un comité de estandarización, por lo tanto permite la comunicación entre equipos de diferentes marcas sin la necesidad de una pasarela de protocolo.

- Independiente de fabricantes, los equipos para *PROFIBÚS* son ofrecidos por muchos vendedores los cuales han de estar certificados.
- Transmisión digital. La comunicación bidireccional entre sistemas de control de procesos y dispositivos de campo es soportada a través de par trenzado, de forma habitual.
- Exactitud, gracias al reconocimiento de comandos y mensajes, *PROFIBÚS* es un sistema de comunicación altamente seguro puesto que los mensajes defectuosos son repetidos hasta que la confirmación de recepción es enviada.
- Multi-funcional, *PROFIBÚS* se adapta a todas las tareas de automatización, permitiendo el intercambio de datos entre controladores como entre elementos de campo.
- De probada funcionalidad.
- Capacidad de diagnóstico. El estándar *PROFIBÚS* define múltiples formas de diagnósticos entre el sistema de control de procesos y los dispositivos de campo.
- Expansión del sistema. Un equipo adicional puede ser incorporado en cualquier momento al bus sin necesidad de reformar la estructura existente, incluso sin enturbiar la comunicación existente.
- Bajo coste. Reduce cableado y simplifica en consecuencia los planos de estos, competencia de precios entre vendedores al ser independiente, etc.
- En constante renovación gracias a *PROFIBÚS International (PI)*.

Como característica fundamental del bus resaltaremos la comunicación abierta. La comunicación abierta se refiere al intercambio de datos entre estaciones diferentes de producción usando diferentes redes de datos estándar.

Profibus tiene, conforme al estándar, cinco diferentes tecnologías de transmisión, que son identificadas como:

- RS-485. Utiliza un par de cobre trenzado apantallado, y permite velocidades entre 9.6 kbit/s y 12 Mbit/s. Hasta 32 estaciones, o más si se utilizan repetidores.
- MBP. Manchester Coding y Bus Powered, es transmisión sincrónica con una velocidad fija de 31.25 kbit/s.
- RS-485 IS. Las versiones IS son intrínsecamente seguras, utilizadas en zonas peligrosas (explosivas).
- MBP IS
- Fibra óptica. Incluye versiones de fibra de vidrio multimodo y monomodo, fibra plástica y fibra HCS.

## **PROFIBÚS DP**

(*Decentralized Periphery*), para el control distribuido. Diseñado para la comunicación entre sistemas de control automático y entradas y salidas distribuidas o remotas en campo. Ofrece la funcionalidad de intercambiar datos de forma rápida y cíclica. Su principal ventaja es que es *plug and play*, en cuanto a que se permite la identificación de los dispositivos.

Es el más utilizado para solucionar las necesidades de interconexión de los posibles perfiles *PROFIBÚS*. Está optimizado en velocidad, eficiencia y bajo costos de conexión, orientado especialmente para la comunicación entre sistemas automáticos y los periféricos distribuidos en el nivel de campo. Cuando se selecciona un medio de transmisión se deben considerar las largas distancias, la velocidad de transmisión y los requerimientos del sistema a automatizar, tales como la operatividad en áreas peligrosas y la transmisión en un único cable de los datos y la energía.

El intercambio de datos es principalmente cíclico, utilizándose determinadas funciones de comunicación eventualmente regladas según EN 50170 a parte de las habituales básicas funciones exclusivas para dicho intercambio. Aunque *DP* también ofrece servicio en comunicaciones acíclicas más complejas para la parametrización, la monitorización y el manejo de alarmas en los dispositivos de campos inteligentes.

## **PROFIBÚS PA**

(*Process Automation*). Además de las definiciones anteriormente tratadas de *PROFIBÚS DP*, *PA* también tiene las suyas propias, tales como tipo de dato y unidad de los datos de medida, significado de los campos del registro estado...

El perfil está basado en los perfiles de comunicación *DP* y dependen del campo de la aplicación, *IEC 1158-2*, *RS-485* y fibra óptica son utilizados como transmisores de esta tecnología. Los perfiles *PA* definen los parámetros de los dispositivos y el comportamiento de los típicos dispositivos de campo tales como transductores de señal o actuadores, facilitando incluso el cambio de un dispositivo por otro similar de distinto fabricante.

La descripción de las funciones y del comportamiento del dispositivo está basado funciones bloques internacionales. La definición y las opciones del perfil de aplicaciones *PA* hacen a *PROFIBÚS* el mejor sustituto para las convencionales líneas de transmisión en 4-20 mA. *PROFIBÚS* también permite controlar la adquisición de datos y la actuación en el medio, en procesos a través de un simple cable de 2 hilos. *PROFIBÚS* permite la conexión y desconexión de dispositivos durante el funcionamiento incluso en áreas peligrosas. Especialmente diseñado para este tipo de áreas peligrosas se ha elaborado el perfil *PROFIBÚS PA*. Ha sido desarrollado en cooperación con usuarios de procesos industriales, con las siguientes características:

- Perfil de aplicaciones estándar para procesos de automatización que permite la intercambiabilidad del dispositivo de campo con otros de distinto fabricante.
- Se pueden añadir y desconectar estaciones del bus incluso en áreas peligrosas sin influenciar a otras estaciones.
- El bus suministra alimentación para los transductores de señal usando la tecnología de dos cables acuerdo con el estándar *IEC 1158-2*.
- Este bus es usado también en áreas potencialmente explosivas con los tipos de protección EEx ia/ib o la encapsulación EEx d.

## 4.5. Software necesario.

Una vez que ya sabemos cómo está conectado todo nuestro sistema, vamos a conocer el software necesario para que todo funcione. Podríamos hacer tres divisiones de nuestro sistema físico según la necesidad de software. La parte de configuración, la parte de programación y la parte de control y supervisión. Para cada una de las partes hemos utilizado un programa distinto.

En primer lugar, nos encargamos de configurar el sensor inteligente *SITRANS Probe LU*, utilizando para ello, el software *PDM*. A continuación, programamos el autómata con el código necesario para el control del sistema físico con *Step 7*. Y por último, realizamos la interfaz de supervisión y control con el programa *WinCC flexible*. Todo el software utilizado está desarrollado por *Siemens AG*.

### 4.5.1. Versiones instaladas.

Un aspecto fundamental a la hora de utilizar tres tipos de software distintos, es tener en cuenta las versiones de cada uno de ellos. La versión de cada uno va a determinar con que otros programas son compatibles y si lo son con nuestro sistema operativo. Por lo tanto, antes de instalar cualquier programa, deberemos consultar la hoja de compatibilidades (ver anexos). Las versiones que hemos utilizado, todas ellas compatibles entre sí y con nuestro sistema operativo son las siguientes:

- Simatic PDM v. 6.0
- Step 7 v. 5.5 service pack 2
- WinCC flexible 2008 service pack 3

#### Simatic PDM.

Como ya hemos comentado con anterioridad, este software nos permite configurar y calibrar el sensor inteligente. Una de las ventajas de este programa es que se puede utilizar con más dispositivos, aportando un amplio catálogo de unidades. En su conjunto, el software se compone de tres discos:

- El primer disco contiene el programa propiamente dicho, con el ejecutable para su instalación.
- El segundo disco contiene información relacionada con el software; guías, manuales, licencias...
- El tercer compacto contiene un amplio catálogo de sensores del fabricante y ha de instalarse para poder acceder a posteriori al dispositivo *SITRANS*.

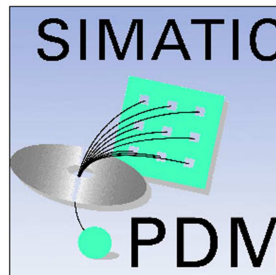


Figura 12: Logo Simatic PDM

### Step 7.

Para configurar el programa que actuará sobre el sistema físico, hemos utilizado el software *Step 7* en su versión indicada anteriormente. Este programa nos permite configurar prácticamente cualquier *CPU*, además de dotarnos de una herramienta muy potente capaz de realizar funciones de control, funciones de conteo, crear alarmas, etc. Además podemos programar el autómatas en tres lenguajes distintos:

- KOP: diagrama de contactos. Sencillo e intuitivo.
- FUP: lenguaje de funciones. Usa funciones para programar. Algo más complejo que el primero.
- AWL: lenguaje de instrucciones. El más complejo de los tres pero a su vez, el que menos código necesita, traducándose eso en menos memoria usada para un mismo programa.

Consta de un disco que contiene el ejecutable del programa, así como documentación relativa al programa; licencias, manuales, guías, conexionado interno...

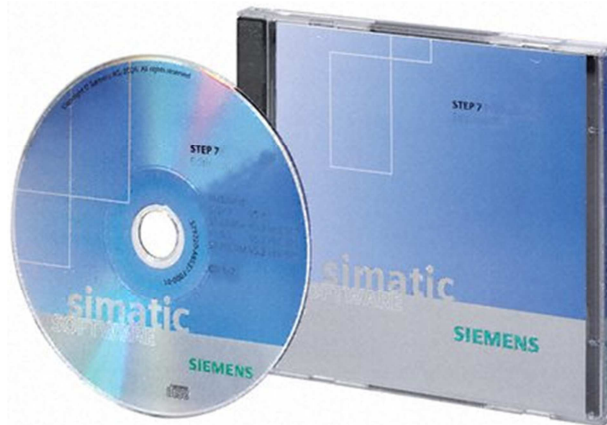


Figura 13: Disco de Step 7

### WinCC flexible 2008.

Por último, para configurar la interfaz hombre-máquina para supervisar y controlar todo el sistema físico, hemos utilizado el software *WinCC flexible 2008*. Es importante tener en cuenta que este software no es el software dedicado a la configuración de sistemas *SCADA* de *Siemens*. La herramienta para dicha tarea es *WinCC*. Aún así con *WinCC flexible* podemos configurar un sistema de control, supervisión y adquisición de datos. El programa viene en un dvd con el service pack 2. Es necesario actualizarlo con el service pack 3 para no afectar a las compatibilidades.

Bastante intuitivo a la hora de configurar las interfaces, contiene también la herramienta de simulación *RunTime* con la que podremos supervisar y controlar todo el proceso.



Figura 14: WinCC flexible 2008

#### 4.5.2. Integración de programas.

Una de las principales características que comparten los programas utilizados, es la capacidad de integración entre ellos. De ahí la importancia en la compatibilidad de los mismos. Con la función de integración es como si tuviéramos tres programas en uno sólo. Pero si queremos llegar a disfrutar de esta ventaja, es necesario seguir un orden a la hora de instalar el software. Para ello, debemos instalar los programas en el orden siguiente:

- Simatic PDM
- Step 7
- WinCC flexible 2008

Es necesario seguir este orden. Es posible integrar *Step 7* y *WinCC flexible* siempre y cuando, pese a no instalar *Simatic PDM*, mantengamos la jerarquía establecida. Podemos ver los programas integrados en la siguiente imagen (Figura 15):

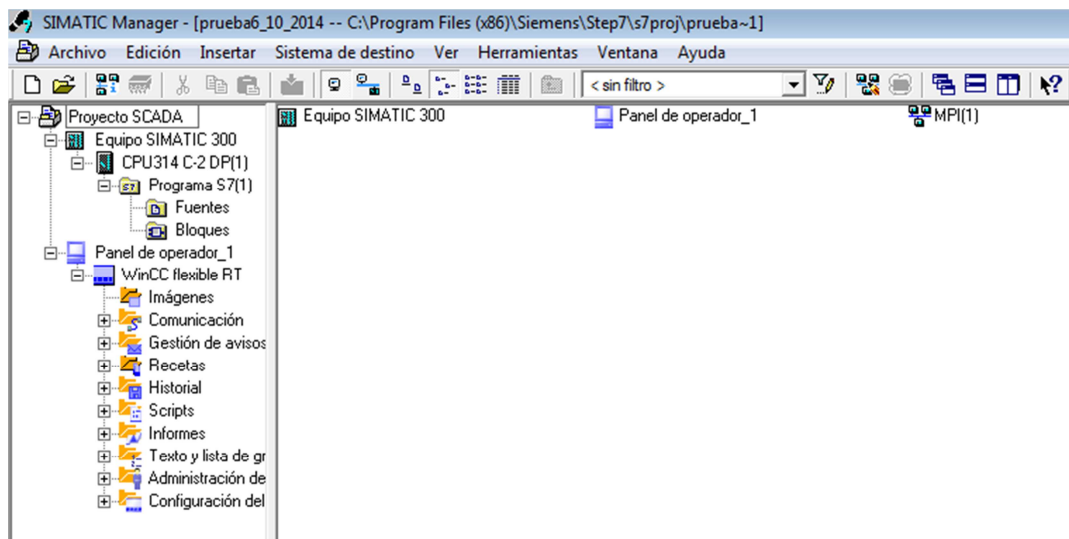


Figura 15: Integración que muestra Step 7 y WinCC flexible juntos

Si se desea conocer más sobre la instalación del software, así como los manuales de los mismos y más información relativa, es aconsejable consultar los anexos.

# Capítulo 5:

## SITRANS Probe LU.

---

Como ya se ha apuntado anteriormente, una pieza importante de este *PFC* es la incorporación del sensor inteligente *SITRANS Probe LU* de *Siemens*. Con este dispositivo ampliamos así la parte docente al proveer al alumno con conocimientos básicos sobre estos dispositivos y sus características. Pero antes de abordar este sensor debemos tener claros algunos conceptos que se explican a continuación.

### 5.1. Introducción a los sensores inteligentes.

El término “*Smart Sensor*” o “*Sensor Inteligente*” se ha utilizado por varios investigadores en varios contextos diferentes; desde sensores que incorporan unos pocos dispositivos activos para proporcionar una señal de calidad, a sensores integrados que incorporan un bloque sofisticado de circuito electrónico con parte analógica y digital que permite convertir un sensor pasivo en un sensor inteligente.

Desde el primer y muy sencillo sensor desarrollado por *Honeywell* en el año 1969, hasta los dispositivos complejos actuales de alta tecnología, estos dispositivos integran muchas funciones automáticas como son: identificación, calibración, comprobación, etc... que permiten obtener sensores que no sólo entregan una señal digital, sino además, linealizada, calibrada, robusta y compatible con otros dispositivos.

La revolución que supuso la aparición de los ordenadores y la miniaturización de componentes basados en semiconductores, hasta la aparición del micro-controlador, ha influido también en el mundo del control de procesos, así se pasó del control centralizado al control distribuido, donde cada vez tiene más importancia los sensores que ofrecen más posibilidades además de la mera obtención de una señal eléctrica a partir de una física.

El desarrollo de estos revolucionarios componentes ha permitido aumentar la eficiencia, calidad y velocidad de los procesos industriales, la investigación y el desarrollo científico. Por lo tanto, se puede definir al sensor inteligente como aquel dispositivo capaz de:

- Acondicionar la señal analógica.
- Conversión A/D.
- Procesamiento primario de la información (validación, linealización, etc)
- Conversión de unidades.
- Comunicarse con la red digital.
- Realizar un diagnóstico de sí mismo (auto-diagnóstico).



En la *Figura 1* podemos ver la estructura básica de los sensores inteligentes en forma de esquema o diagrama de bloques.

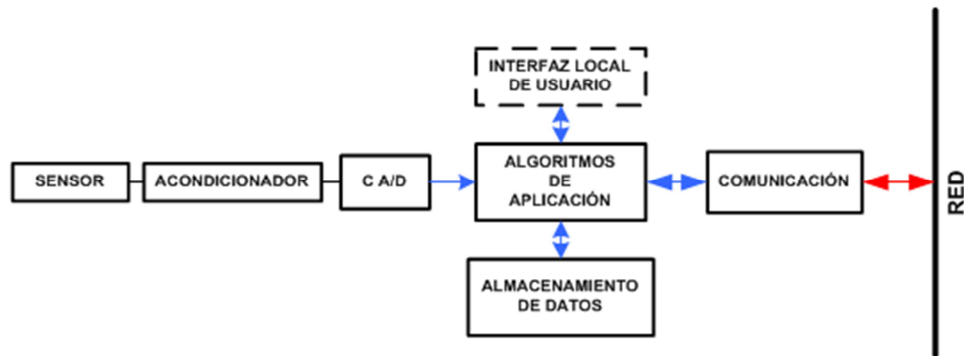


Figura 1: Diagrama con la estructura básica de un sensor inteligente

En la década de los 80, con el desarrollo de los microprocesadores para el procesamiento digital de señales (*DSP*), varias compañías desarrollaron circuitos integrados con la arquitectura de la *Figura 2*, constituyendo los primeros transmisores inteligentes.

Estaban constituidos por varios sensores multiplexados y posteriormente se procesaban las señales, obteniendo una señal de salida para un bucle de corriente. Este funcionamiento dista de la definición que hemos hecho anteriormente del sensor inteligente, pero era el primer paso para llegar a los dispositivos que tenemos hoy día.

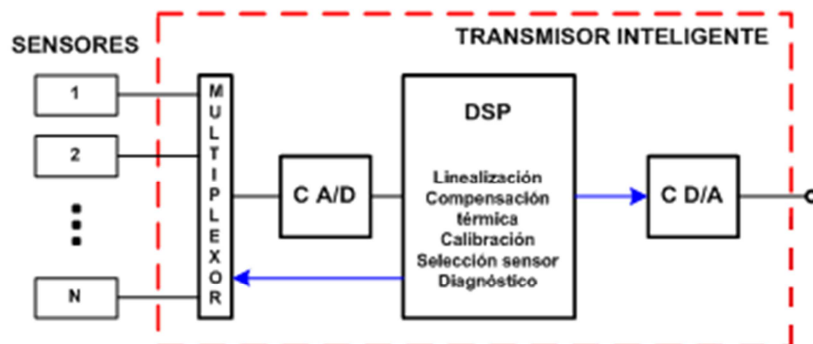


Figura 2: Arquitectura de los primeros transmisores inteligentes

Posteriormente, con el avance en la integración de circuitos se desarrollaron los *MEMS* (*Micro ElectroMechanical Systems*), en los que se integra el propio sensor en el silicio. En estos sistemas se desarrolla el sensor, el acondicionamiento de señal en el mismo bloque y los más avanzados incluyen un micro-controlador con conversión *A/D*, *D/A* y comunicaciones con un bus.

Como ejemplo podemos observar la *Figura 3* donde se muestra un sensor capacitivo de presión diferencial en tan sólo 9 mm<sup>2</sup>. Posee un circuito de acondicionamiento con capacidad de programación y un circuito de compensación de temperatura.

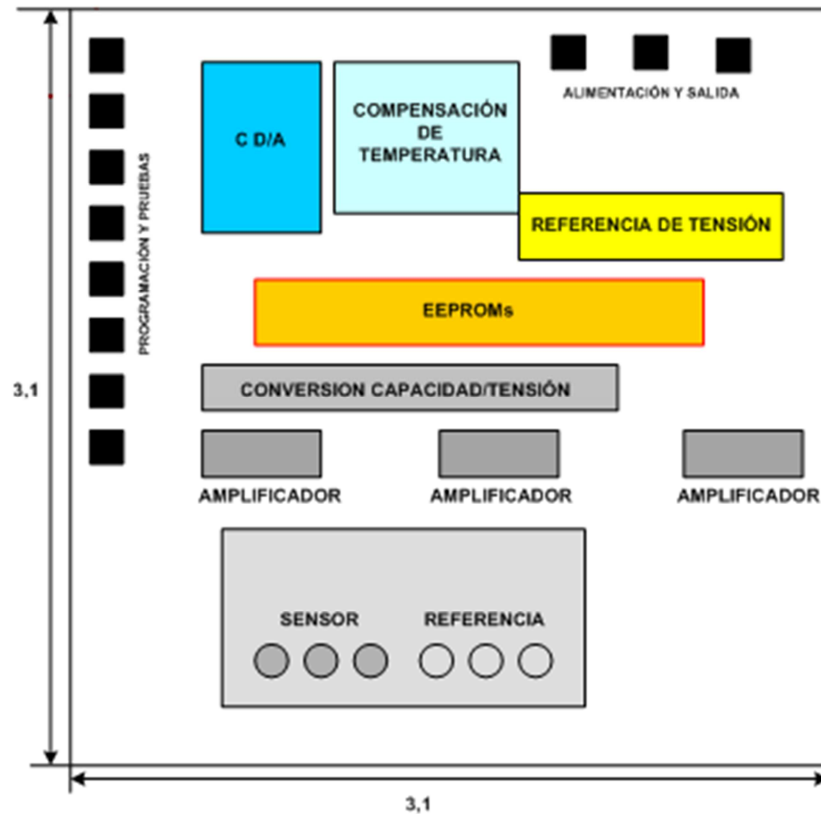


Figura 3: Sensor de presión como ejemplo de un micro sistema electro-mecánico

Por sólo citar algunos ejemplos, los *MEMS* se utilizan en la industria automovilística, tanto en la robótica que los ensambla como en los sistemas de los frenos, encendido automático de luces y limpiaparabrisas, *airbag*, caja de cambios y en el motor de los coches. En nuestras casas se encuentran presentes en la calefacción, ventilación y aire acondicionado, para la regulación de la presión, humedad y temperatura.

Para finalizar con esta introducción, es necesario hablar también del estándar *IEEE-P1451* que regula los sensores inteligentes y las comunicaciones con el bus. Este estándar define a los sensores inteligentes como un sistema conectado a una red común, con la información necesaria para saber si es un sensor o un actuador. De esta forma, se divide el sensor inteligente en dos partes:

- *STIM*.

y

- *NCAP*.

La parte *STIM* contiene al sensor o sensores, el acondicionamiento, la conversión *A/D*, un puerto de intercambio de información con *NCAP* y una tabla de memoria *TEDS* (*Transducer Electronic Data Sheet*), que almacena toda la información de los sensores, fabricante, que mide, rango de la medida, datos de calibración, precisión... Mientras, por su parte, la zona *NCAP* es un procesador que administra las comunicaciones con la red. En la *Figura 4* se observan las dos partes y sus zonas de funcionamiento.

El funcionamiento es sencillo. La parte *STIM* entrega a la parte *NCAP* la medida realizada y todos los datos del sensor que se encuentran en la tabla de memoria y el *NCAP* se encarga de normalizarlos para enviarlos a la red.

Las tablas de memoria, *TEDS*, pueden encontrarse en el propio *STIM* o pueden ser virtuales y estar en una base de datos que a la que se accederá por la red. De esta manera se pueden seguir usando los sensores tradicionales sin incorporar la memoria dentro del estándar.

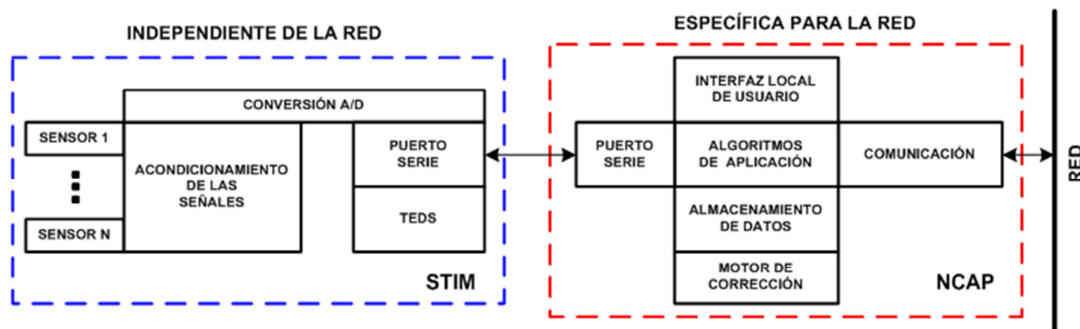


Figura 4: Partes *STIM* y *NCAP* de un sensor inteligente según el estándar IEEE-P1451

## 5.2. Evolución de los sensores inteligentes.

Se pueden distinguir varias etapas o generaciones en la evolución de este tipo de dispositivos desde que apareciera el primer sensor inteligente, allá por 1969. Si se observa la *Figura 5* que representa el diagrama de un sistema de control típico, se puede ver la importancia de los sensores en dicho sistema.

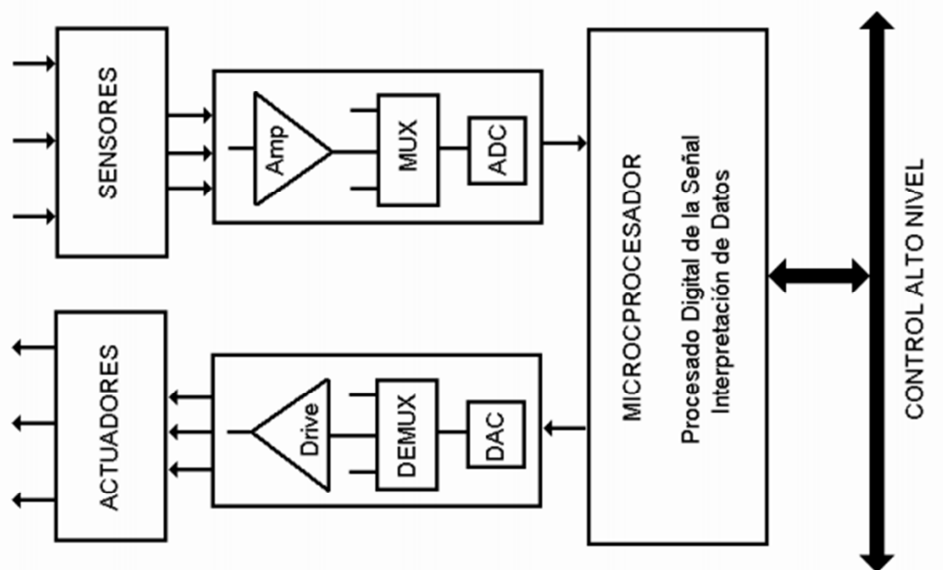


Figura 5: Diagrama de un sistema típico de control

Los sensores proporcionan información analógica al sistema y esta información se acondiciona antes de pasar al microprocesador. El procesador interpreta la información, realiza las actuaciones necesarias y aplican esas decisiones vía los actuadores. Como puede observarse, el sensor es un mero “transmisor”.

Sin embargo, si se presta atención a la *Figura 6* se puede ver un cambio sustancial en el papel del sensor en un sistema de control.

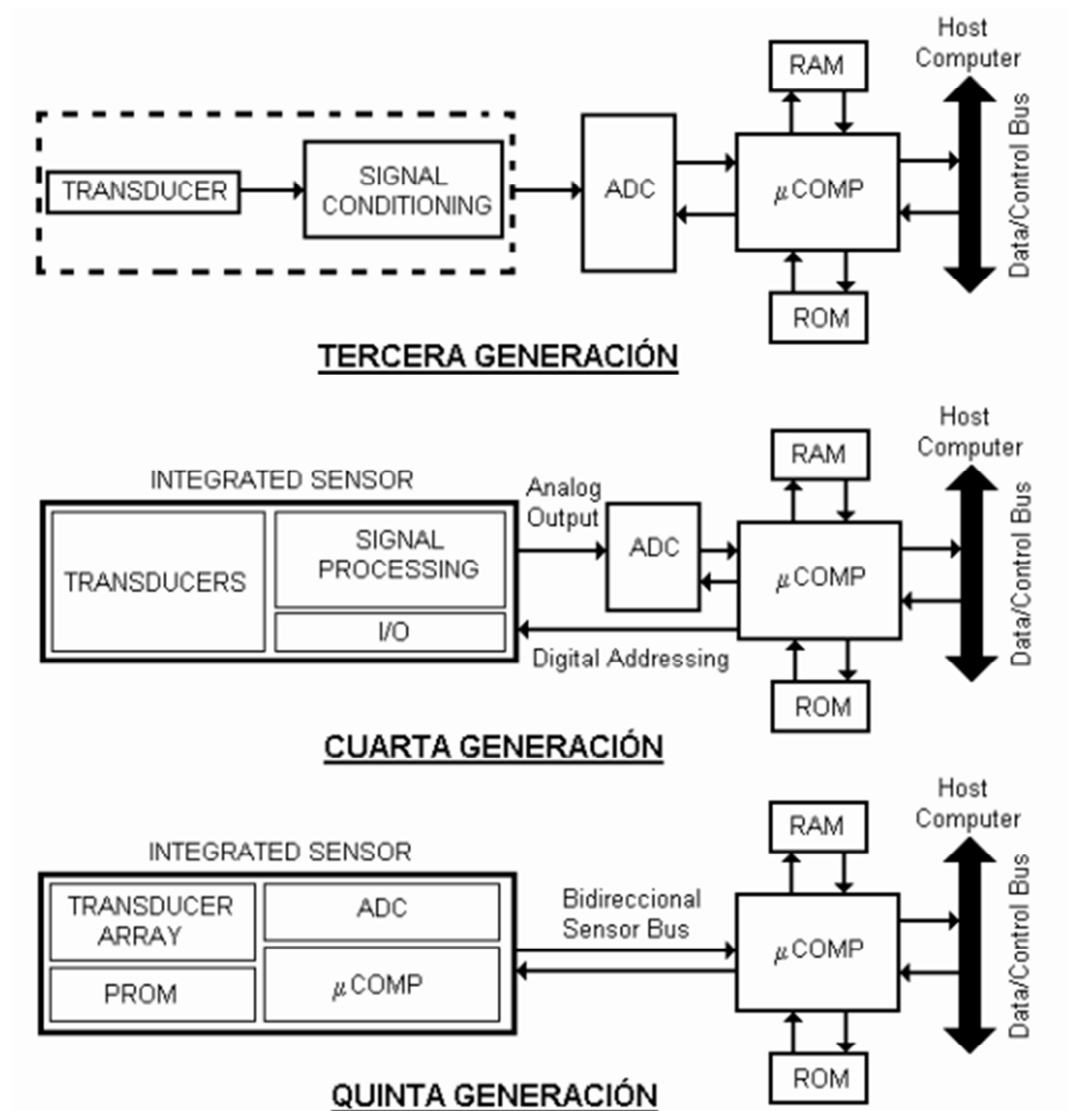


Figura 6: Evolución de los sensores inteligentes

Si se hace un breve repaso sobre las distintas etapas, se puede decir que los dispositivos de primera generación no tienen electrónica asociada, mientras que los sensores de segunda generación forman parte de sistemas puramente analógicos con un control remoto del sensor.

Por otra parte los sensores de tercera generación, en los que se basan la mayoría de los sistemas actuales, realizan la primera etapa de amplificación en el módulo del sensor o en el mismo chip. Así, la señal que aportan los mismos es una señal analógica de alto nivel, codificado, con una variación de tensión o como una señal de frecuencia variable. Esta señal se convierte a digital y posteriormente se procesa mediante un microprocesador.

La cuarta generación de sensores se caracteriza por que la mayor parte de la electrónica, tanto analógica como digital, está en un chip, permitiendo, así, el direccionamiento del sensor y en algunos casos el auto-test mediante comunicación entre el sensor y el micro-controlador.

Por último, los sensores de quinta generación, realizan la conversión de datos en el módulo del sensor, para que la conexión bidireccional entre el micro-controlador sea digital. Estos dispositivos se pueden compensar, digitalmente, utilizando *PROM's*. Este tipo de sensores está caracterizado por varios atributos: comunicación bidireccional de datos y ordenes, transmisión totalmente digital, procesamiento digital local, testeo propio, algoritmos definidos por usuario y algoritmos de compensación.

### 5.3. Arquitectura general de un sensor inteligente.

Para implementar sensores inteligentes se requiere la inclusión de tres partes principales: el procesamiento de la señal, el control y manipulación de las señales digitales y la comunicación con el exterior mediante un bus. A continuación se explicarán con más detalle cada una de las tres.

#### 5.3.1. Procesado de señal.

Las señales que registran los sensores normalmente son bajas en amplitud y la interfaz del sensor presenta una alta impedancia en las frecuencias de funcionamiento normal. La integración de la electrónica de la interfaz y los circuitos que procesan la señal, dentro del mismo sensor, tienen las siguientes funciones: la amplificación de la señal, la transformación de la impedancia, el filtrado de señal, los buffers y el multiplexado.

La amplificación de la señal del sensor es una de las funciones más importantes del sensor para muchas aplicaciones, sobre todo para aquellas en las que la señal es típicamente baja. La ampliación de estas señales en el chip no sólo aumenta la relación señal/ruido sino que reduce el efecto del ruido exterior y además permite aprovechar el máximo rango del convertidor analógico-digital, antes de enviar la señal al exterior. Esta amplificación se puede conseguir utilizando amplificadores *MOS* y bipolares. También se puede conseguir utilizando amplificadores *CMOS*, los cuales son mejores para obtener alta ganancia e impedancia. Estos amplificadores son de 3 a 5 veces más pequeños que los bipolares, por lo que permite incluir decenas de estos amplificadores por canal en el chip. Los amplificadores más sofisticados incorporan la ganancia programable con el objeto para aumentar la relación señal/ruido y para ajustar el rango de salida del amplificador con el rango del convertidor analógico-digital.

En estos amplificadores, también, se puede incorporar el filtrado de señal. En los sistemas multisensor y en los que hay un multiplexor, a menudo se requiere un filtrado de la señal para prevenir el problema de la aliasing que puede introducir un ruido de alta frecuencia. El filtrado de señal mejora también la proporción señal/ruido, debido a que filtra el ruido de alta frecuencia que puede ser introducido por los propios dispositivos electrónicos.

Otra función importante para la mayoría de sensores es la reducción de canales de salida. La multiplexión de datos reduce la cantidad de circuitos requeridos para el sensor, por ejemplo compartiendo el *ADC* entre varios sensores.

Como puede verse, ésta es una de las partes más importantes que conforman cualquier sensor inteligente. Adecuar bien la señal de entrada permitirá obtener una mejor lectura, por lo tanto, cuanto mejor sea el procesamiento de la señal, más preciso será el aparato.

### 5.3.2. Control digital.

La compatibilidad de los sensores inteligentes con los sistemas digitales y basados en microprocesadores es uno de los puntos más importantes. La mayoría de sensores de alto rendimiento deben proporcionar una señal digital para poder acceder a un bus del mismo tipo. Una vez digitalizados los datos captados por el sensor, se puede realizar una variedad de procesamiento de señal para corregir varios errores. Estos incluyen la corrección de la desviación, una auto calibración, detección y corrección de faltas y la corrección de la linealidad.

Para poder manipular la señal del sensor se necesita como elemento principal el convertidor analógico-digital. Estos circuitos han experimentado un gran progreso durante los años precedentes. Las técnicas utilizadas han permitido la integración de funciones analógicas de alto rendimiento en circuitos digitales *CMOS* de alta densidad y baja potencia, permitiendo así la implementación de *ADC's* de alta precisión. Para la mayoría de las aplicaciones las tasas de conversión desde 10 hasta 20 kHz a 12 bits es adecuado. Para algunas aplicaciones puede ser necesario añadir, antes del *ADC*, un amplificador de ganancia programable con el objetivo de aprovechar al máximo el rango de entrada del convertidor. Hay que tener en cuenta que para las aplicaciones del sensor es importante mantener el diseño general tan sencillo como sea posible y ocupar así el menor espacio posible. Por lo tanto, el diseño del convertidor, para aplicaciones de propósito general puede requerir otro conjunto de prioridades. Puede que sea más interesante mejorar la precisión del convertidor a cambio de compensar externamente los errores del convertidor. El diseño del *ADC*, para diferentes aplicaciones, dependerá de los requisitos de la aplicación y de la complejidad del proceso de fabricación.

Una vez digitalizada la señal del sensor se pueden realizar varias funciones. La auto-calibración es una función muy interesante para estos dispositivos. La mayoría de los sensores se deben ajustar tanto en la ganancia como en la desviación, en fábrica y antes de embalar. El funcionamiento de muchos sensores depende, en gran medida, de estos dos parámetros y, por lo tanto, la precisión dependerá de este ajuste. Además, la estabilidad del sensor se ve afectada en tanto en cuanto estos parámetros evolucionen a lo largo del tiempo modificando, así, la precisión del sensor. Por lo tanto es deseable no sólo poder calibrar en fábrica sino también calibrarlo en campo. Para la calibración de cualquier circuito se necesita otro circuito adicional con su complejidad. Esto se puede conseguir tanto con circuitos analógicos como digitales. Para evitar complicaciones y para mejorar el rendimiento general, se puede calibrar utilizando un procesador externo con acceso al *PROM* y por lo tanto a las características del sensor como: la dirección, tipo de sensor, coeficientes del polinomio que serán utilizados para compensar digitalmente el sensor.

Otra de las características principales es la compensación de los datos del sensor inteligente. La compensación de datos se puede utilizar para corregir características como la sensibilidad, la falta de linealidad o la variación de la señal con el tiempo y el ruido. Esto nos lleva a obtener un sensor más preciso y fiable.

Acerca de las funciones que se pueden implementar en el sensor están las *DSP*. Estas técnicas de *DSP*, se pueden añadir para mejorar el funcionamiento del sensor y además posee una característica de computación. El sensor, ahora, es capaz de realizar funciones lógicas y de informar al controlador en caso de un error. El poder computacional que poseen se puede utilizar para modificar automáticamente el rango de los amplificadores, del *ADC*, etc.

Además de la calibración y de la compensación, el auto-test y el diagnóstico son dos funciones necesarias para estos aparatos. La capacidad de auto-test es muy importante ya que permite conocer la funcionalidad del sensor sin tener que quitarlo de su lugar de funcionamiento. El auto-test se puede iniciar bajo el control externo, como por ejemplo de un controlador, para asegurar el buen funcionamiento del sensor. El auto-test es, únicamente para detectar el funcionamiento defectuoso y no se puede utilizar para determinar la calibración exacta.

Otra característica importante para varios sistemas de control y de instrumentación es la fiabilidad. Esto es, especialmente importante, en sistemas distribuidos donde el acceso a un determinado sensor no es sencillo y, por lo tanto es necesario que sea extremadamente seguro. Los sensores inteligentes proporcionan una clara ventaja con respecto a los sensores pasivos. El método más sencillo para mejorar la fiabilidad es agregando redundancia al sensor. Esto es posible debido a su pequeño tamaño, por lo que se puede implementar una réplica del sensor en el mismo chip. También es posible duplicar los bloques asociados al sensor y situarlos en el chip para aumentar el rendimiento general. Esta redundancia no afecta al procesamiento de señal, la única desventaja viene por la limitación de espacio.

Otra posibilidad para mejorar el rendimiento y la fiabilidad sería minimizando el número de bloques en el chip. De este modo la posibilidad de fallo del dispositivo se reduce. Se deduce, así, que debe existir un compromiso entre el nivel de complejidad, y por lo tanto de la funcionalidad, y el costo general y el rendimiento.

### 5.3.3. Comunicación.

La última función importante que debe desempeñar un sensor inteligente es la comunicación con el resto de componentes del sistema de control, especialmente con los controladores, de tal forma que una parte del circuito del sensor, debe dedicarse a la interfaz con el bus digital para intercambiar información. Hay que destacar dos puntos importantes:

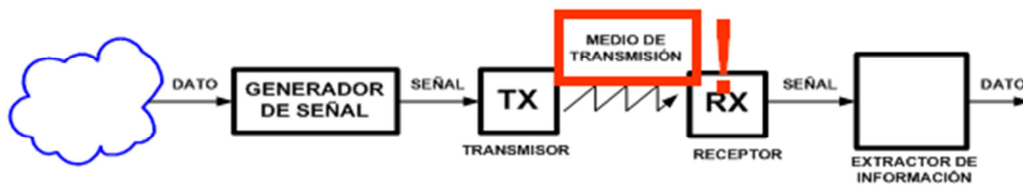
- Cada sensor debe ser capaz de comunicarse mediante varios buses y protocolos de bus, especialmente ante la falta de un bus universalmente aceptado para sistemas de medida.
- Intercambio de información entre el sensor y el controlador a través del bus digital.

La información a intercambiar serán: los datos de calibración, compensación, direccionamiento, información propia del sensor, los datos medidos y también la posibilidad de poder programarlo. En los modelos más complejos, la interfaz de comunicaciones debe realizar el intercambio de datos, a alta velocidad, no solo con el controlador sino también con los sensores que existen en el sistema distribuido. En el próximo punto veremos los sistemas de comunicación de los sensores.

## 5.4. Sistemas de comunicación para sensores.

En algunas ocasiones, el lugar donde se genera la información de las variables de un sistema de instrumentación está separado una cierta distancia del lugar en que esa información va a ser utilizada. Esto sucede normalmente en las plantas industriales, donde los procesos de producción están en un edificio y el control se ejerce en otro distinto. En estos casos, es necesario que las señales recorran distancias más o menos grandes en un proceso genérico que se denomina transmisión de información.

Dentro de esta transmisión de información debemos distinguir entre datos y señal. Los datos conforman la parte que contiene la información y la señal es la forma en la que se configuran los datos para ser transmitidos o enviados. También hay que reseñar el medio de transmisión de la información. En un lenguaje más común, sería el camino que utiliza la información para llegar desde del proceso al puesto de control y viceversa. En la *Figura 7* se muestra el esquema típico de la transmisión de información.



*Figura 7: Transmisión de información*

Durante este “camino”, la señal puede sufrir perturbaciones. El medio de transmisión debe permitir que la señal se transmita sin atenuarla, distorsionarla o producir ruido. Podemos distinguir entre:

- Medios guiados: suponen el confinamiento de las señales en el medio por el cual se propagan. Ofrecen una relativa alta inmunidad de las señales que se propagan por ellos a las interferencias exteriores; sin embargo, la necesidad de un nexo físico entre emisor y receptor limita su aplicación a sistemas sin movilidad.
- Medios no guiados: el soporte es ilimitado o cuasi-ilimitado y las señales evolucionan “libremente” por el medio (aire, agua o vacío).

En la siguiente imagen (*Figura 8*) podemos ver representados ambos medios de transmisión de señales.



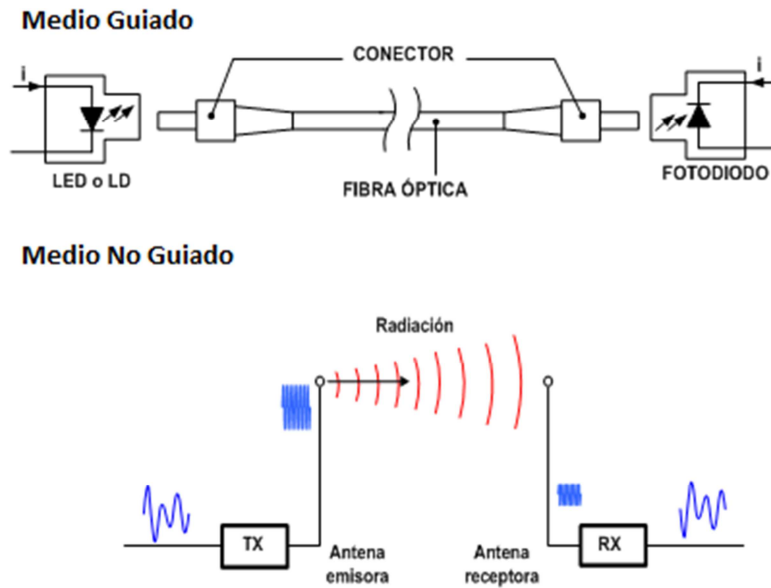


Figura 8: Medios guiado y no guiado de transmisión

#### 5.4.1. Protocolo HART.

El protocolo de comunicación *HART* (*Highway Addressable Remote Transducer*) es un protocolo estándar en la industria para la comunicación de instrumentos de campo inteligentes basados en microprocesador. Dado que existen muchas empresas que tienen sus sistemas cableados 4-20 mA, la sustitución directa de sistemas analógicos por sistemas digitales sería excesivamente cara. Para dotar de capacidad de comunicación digital a los sistemas analógicos existentes, la empresa *Rosemount* propuso este protocolo.

El protocolo *HART* permite la superposición de señal de comunicación digital a las señales analógicas de 4-20 mA, sin interferencia, en el mismo cableado. Proporciona también algunos de los beneficios dados por la tecnología *Fieldbus*, manteniendo la compatibilidad con la instrumentación analógica y aprovechando el conocimiento ya dominado sobre los sistemas 4-20 mA existentes.

La posibilidad de comunicación digital permite tener toda la información sobre un dispositivo en el propio instrumento (fabricante, modelo, número de serie, factores de calibración, límites de medida, etc). Esta información es esencial en la puesta en marcha de la instalación y reduce los costes de mantenimiento.

Debemos añadir también que el protocolo *HART* permite la comunicación digital bidireccional en instrumentos de campo inteligentes sin interferir en la señal analógica de 4-20 mA. Tanto la señal analógica como la señal digital de comunicación *HART*, pueden ser transmitidas simultáneamente en el mismo par de hilos, aunque lo mejor sea mantener ambas señales separadas. La variable primaria es la información de la señal de control que puede ser transmitida por la señal de 4-20 mA, mientras que las mediciones adicionales, parámetros de proceso, configuración del instrumento, calibración y las informaciones de diagnóstico están disponibles en el mismo par de hilos y al mismo tiempo. A diferencia de otras tecnologías de comunicación digitales abiertas para instrumentación de procesos, el protocolo *HART* es compatible con los sistemas existentes. En la *Figura 9* se muestra un ejemplo representativo de dicho protocolo de comunicación:

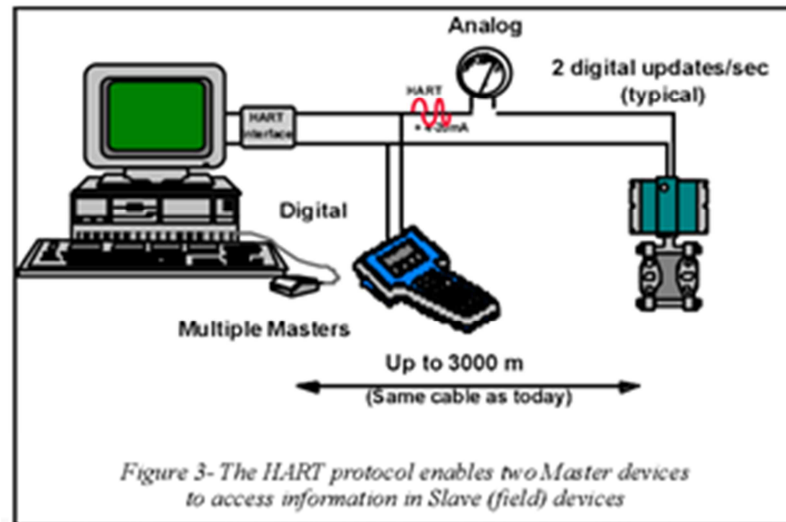


Figura 9: Protocolo HART

Tal y como muestra la imagen anterior (*Figura 9*), *HART* es un protocolo del tipo maestro/esclavo, lo que significa que un instrumento de campo (esclavo) solamente “responde” cuando sea “preguntado” por un maestro. Dos maestros (primario y secundario) se pueden comunicar con un instrumento esclavo en una red *HART*. Los maestros secundarios, como los configuradores portátiles pueden ser conectados normalmente en cualquier punto de la red y comunicarse con los instrumentos de campo sin provocar problemas en la comunicación con el maestro primario.

El protocolo *HART* emplea un sistema de modulación estándar (*BELL 202*) que es compatible con los pares trenzados ya instalados. Se puede emplear tanto un solo par trenzado de hasta 3000 m, como múltiples pares con un blindaje común, de hasta 1500 m de longitud. La modulación por cambios de frecuencia (*FSK*) se usa para superponer las señales de comunicación digital a la señal de 4-20 mA. La señal *HART FSK* permite la comunicación digital en dos direcciones, haciendo posible la transmisión y recepción para obtener información adicional. El protocolo *HART* se propaga a una velocidad de 1200 bits por segundo, sin interrumpir la señal de 4-20 mA, representando un 0 lógico con una frecuencia de 2200 Hz y un 1 lógico con 1200 Hz tal y como puede verse en la *Figura 10*.

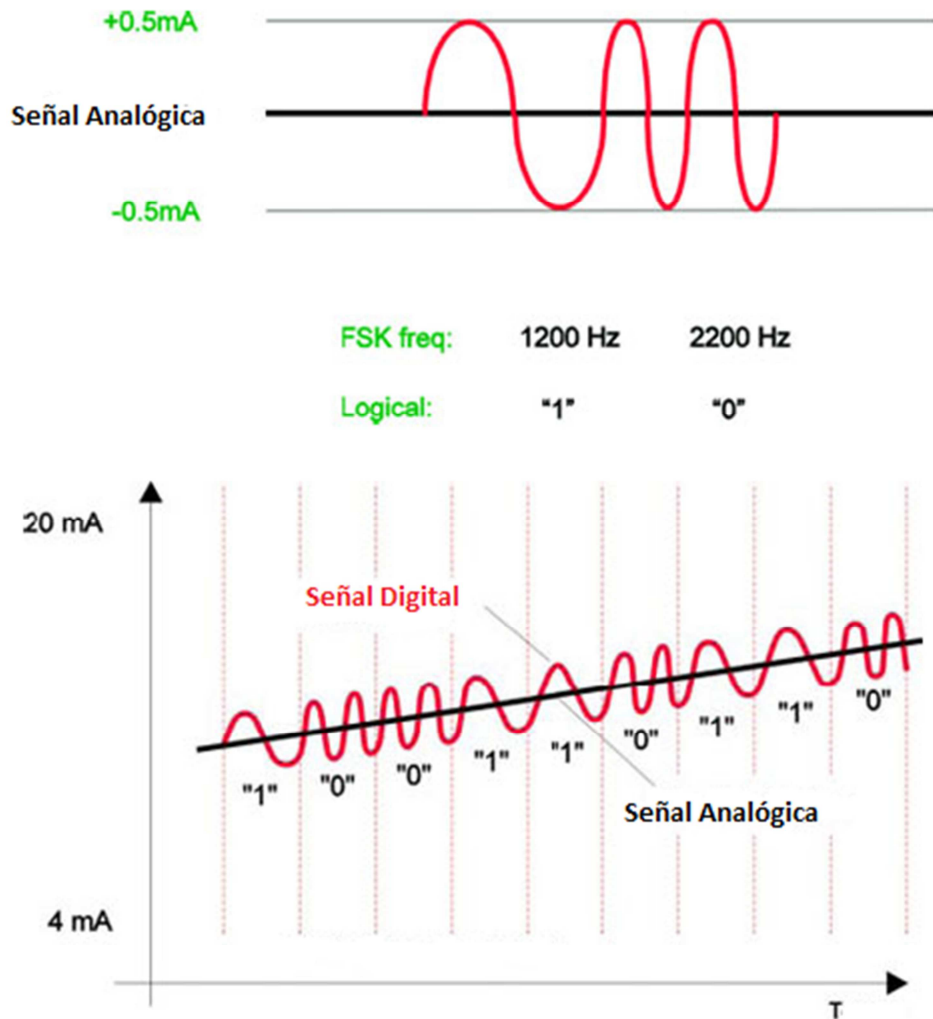


Figura 10: Modulación por cambios de frecuencia FSK

#### 5.4.2. El bus de campo.

Los sistemas de 4-20 mA presentan un problema. Si la salida del sensor es analógica, para linealizarla, detectar límites u otras operaciones, puede ser necesario digitalizarla para que la procese un micro-controlador. Si luego se debe transmitir con un bucle 4-20 mA, hay que convertirla de nuevo en analógica, y convertirla otra vez en digital en el procesador central.

Si además la comunicación es tipo bus, ya no se emplea un canal para una única información, sino que el mismo canal físico es compartido por varias informaciones distintas, viajando en los dos sentidos permitiendo así un ahorro de cableado.

La incorporación de nuevos sensores y actuadores en un proceso sólo hace falta tenerla en cuenta en el software del sistema, no en el cableado. Además, la instalación de un nuevo transmisor no exige detener el proceso, sino que basta transmitirle los datos de calibración, linealización, alarmas, etc. Pero para que distintos instrumentos sean compatibles con un mismo bus, debe haber un acuerdo en sus especificaciones.

Por tanto, podemos definir al bus de campo como un sistema de dispositivos de campo (sensores y actuadores) y dispositivos de control, que comparten un bus digital serie bidireccional para transmitir informaciones entre ellos, sustituyendo a la convencional transmisión analógica punto a punto.

Además de la reducción de costos, los buses de campo incorporan otras ventajas que se pueden enumerar a continuación:

- Flexibilidad: el montaje de un nuevo instrumento, supone la simple conexión eléctrica al bus y una posterior configuración remota (desde la sala de control). Al tratarse de buses abiertos, resultará posible la conexión de instrumentos de distintos fabricantes a un mismo bus.
- Seguridad: transmisión simultánea de señales de diagnóstico de sensores y actuadores, permitiendo así instalaciones más seguras, ya que permite incluir en el control, actuaciones ante fallos de periféricos.
- Precisión: transmisión totalmente digital (codificación *IEEE 754* 32 bits) para variables analógicas.
- Facilidad de mantenimiento: resulta posible diagnosticar un incorrecto funcionamiento del instrumento, y realizar calibraciones de forma remota desde sala de control. Al sustituir la transmisión analógica por una digital, fue posible desarrollar un protocolo inteligente permitiera funciones distribuidas a los sensores y actuadores, permitiendo la inclusión de reguladores de proceso embebidos en los propios sensores o actuadores.
- Economía: reducción de costos de montaje (cableado) y de costos de mantenimiento.
- Reducción de conductores de unión.
- Eliminación de paneles de conexionado (*Marshalling*).
- Acopladores de segmento en ambientes explosivos.
- Eliminación de módulos de conversión *A/D*.

En la *Figura 11* podemos ver una comparativa entre el conexionado usando buses de campo y el utilizado convencionalmente:

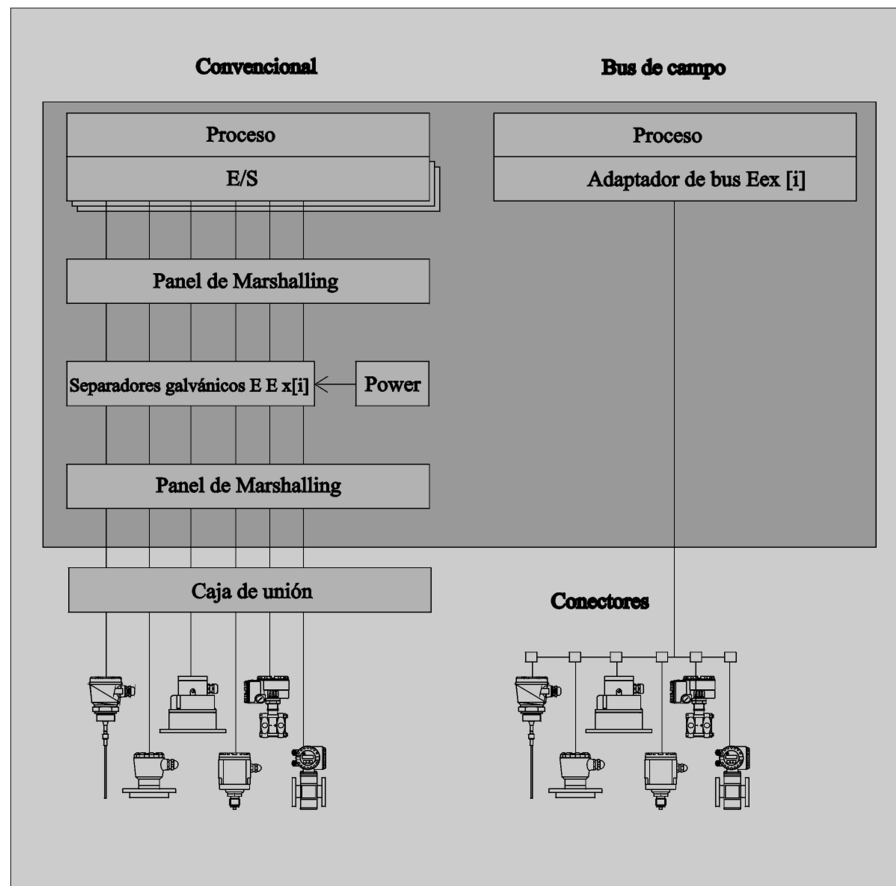


Figura 11: Comparativa de conexiones

Ahora que ya sabemos las ventajas que tenemos con los buses de campo, hay que señalar que no existe un único patrón, existiendo en el mercado los estándares *Fieldbus* y *PROFIBUS*. Además se usa el estándar *ETHERNET* como bus a nivel de área. En Europa, el estándar utilizado es *PROFIBUS* en sus tres perfiles:

- *PROFIBUS FMS*: esta es la solución universal para las tareas de comunicación a nivel de celda. Está diseñado para proveer comunicaciones con controladores y dispositivos de campo inteligentes.
- *PROFIBUS DP*<sup>1</sup>: esta es la versión de *PROFIBUS* dedicada específicamente a comunicaciones de tiempo crítico entre sistemas de automatización y periféricos distribuidos, reemplazando el costoso cableado paralelo de 24 VDC y 4 a 20 mA utilizado en el enlace con campo. *PROFIBUS DP*, limita la cantidad de información a transmitir para lograr una mejor prestación, llegando a sólo 6 ms para la transmisión de 512 bits de información de entradas y salidas distribuidas en 32 estaciones.
- *PROFIBUS PA*<sup>2</sup>: es la versión utilizada en automatización de procesos, utilizando la técnica de transmisión especificada en *IEC 1158-2*, permitiendo seguridad intrínseca y estaciones alimentadas por el bus. Los perfiles de los transmisores se basan en el protocolo *PROFIBUS DP* y utilizan las ampliaciones *DP* que, además de la actualización cíclica de las variables del proceso, permiten un acceso acíclico a los datos de parametrización.

En la siguiente imagen (*Figura 12*) se observa con detalle la utilización de los tres perfiles *PROFIBUS* en un proceso de control.

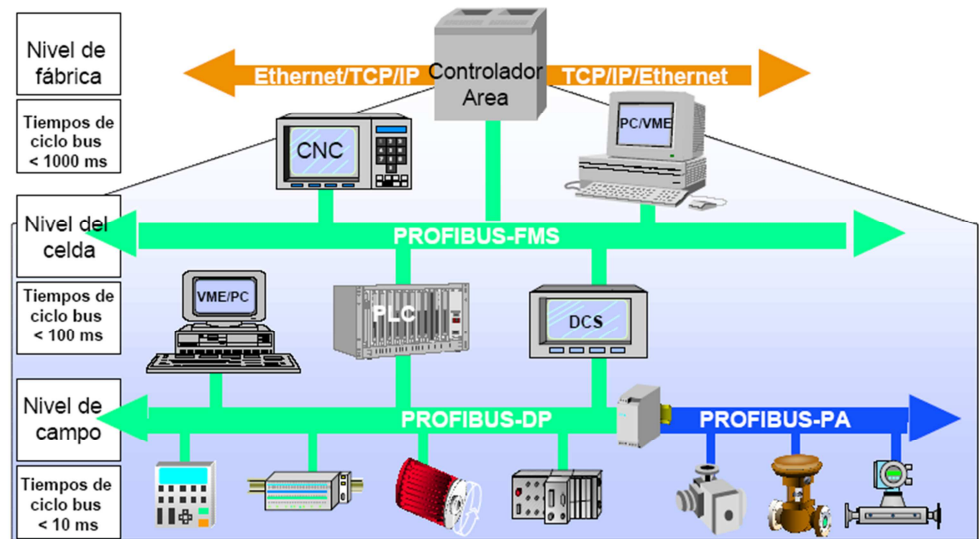


Figura 12: Los tres perfiles PROFIBUS con el estándar ETHERNET

<sup>1) 2)</sup> Ya ha sido explicado con más detalle en el cuarto capítulo de este *PFC*. Simplemente se ha considerado conveniente volver a mencionar de forma breve este punto para que el lector pueda hacer una comparativa entre el protocolo *HART* y los estándares de los buses de campo y sus perfiles.

Por último, para tener una información más completa de los buses de campo, se mostrará una tabla comparativa (*Figura 13*) de los estándares de comunicación.

<b>DeviceNet</b>	Mercado americano Industria de fabricación (automóvil)
<b>Fieldbus Foundation</b>	Mercado americano Industria de proceso (petroquímica ) Zonas clasificadas con riesgo de incendio o explosión
<b>Profibus</b>	Mercado europeo Industria de proceso (Versión PA) Industria de fabricación (Versión DP)
<b>AS-Interface</b>	Comunicación a nivel de campo.
<b>ControlNet</b>	Comunicación de elementos a nivel de célula entre PLC's, SCADA, etc. Comunicación con periferia de E/S descentralizada
<b>Modbus</b>	Industria de fabricación Comunicación con Periferia E/S descentralizada
<b>LonWorks</b>	Industria de fabricación Comunicación con Periferia E/S descentralizada
<b>Industrial Ethernet</b>	Comunicación a nivel de área.
<b>EIB</b>	Bus especialmente aplicado a Domótica

*Figura 13: Tabla comparativa de los buses de campo*

## 5.5. SITRANS Probe LU.

Una vez que ya tenemos claro los conceptos e información necesaria para entender el funcionamiento de un sensor inteligente, es hora de centrarnos en el sensor *SITRANS Probe LU*.

### 5.5.1. Información general.

El dispositivo *SITRANS Probe LU* es un sensor inteligente o “Smart Sensor” de dos hilos, basado en la tecnología de ultrasonidos, lo cual lo convierte en un aparato perfecto para nuestro sistema de control. Desarrollado por *Siemens*, el sensor *SITRANS* es un transmisor ultrasónico para la medición de nivel y de volumen de líquidos en depósitos de almacenamiento y recipientes de proceso simples, así como para la medición de caudal en canales abiertos.

Es un dispositivo ideal para las mediciones de nivel en la industria del agua y aguas residuales, así como para el almacenamiento de productos químicos. Existen dos versiones dependiendo del alcance del sensor, siendo una de seis metros y otra de doce. En este *PFC* se ha trabajado con la primera versión. En la siguiente imagen (*Figura 14*) podemos ver el sensor:



Figura 14: Sensor SITRANS Probe LU

Este aparato está disponible con sensores de *ETFE* (etilenotetrafluoretileno) o *PVDF* (fluoruro de polivinilideno); materiales resistentes al calor, para estar a la altura de los requisitos químicos de su aplicación. Para los procesos con temperaturas de material o de proceso variables, el dispositivo dispone de un sensor de temperatura integrado para compensar los errores de medida por fluctuaciones en la dicha magnitud. Además, el sensor está disponible en tres versiones:

- Uso general para ambientes no peligrosos.
- Intrínsecamente seguro para entornos algo más conflictivos como depósitos abiertos a la intemperie.
- Anti incendio para entornos con un alto riesgo como pueden ser procesos químicos a temperaturas muy elevadas.

Como ya hemos apuntado con anterioridad, el sensor, dependiendo de su versión, tiene un rango máximo de alcance de 6 o 12 metros. Entre otras características<sup>1</sup> podemos citar:

- Rango de medida: de 0.25 a 6 o 12 m. (según versión; 6 metros en nuestro modelo).
- Temperatura de proceso: de -40 a 85 °C
- Señal: de 4 a 20 mA para comunicación *HART*.  
digital para *PROFIBUS PA*.  
seguridad intrínseca para entornos peligrosos (opcional según versión).



Por otro lado, al adquirir el sensor, éste viene acompañado del software de configuración *SIMATIC PDM* y con un controlador remoto desde el que podemos configurar el aparato sin necesidad de ningún programa informático. En la foto (Figura 15) podemos ver control a distancia.



Figura 15: Control remoto

En último lugar, y para tener una idea general de lo que puede ofrecernos este instrumento, vamos a destacar las ventajas del mismo:

- Medición continua del nivel de relleno<sup>2</sup> en rangos de medida de hasta 12 m (alcance máximo según versión).
- Fácil instalación y puesta en marcha.
- Programación con control remoto manual de seguridad intrínseca por infrarrojos.
- Comunicación mediante *HART* o *PROFIBUS PA*.
- Sensores ultrasónicos *ETFE* o *PVDF* para la compatibilidad química.
- Procesamiento de señales patentado *Sonic Intelligence* para mejorar la obtención de resultados.
- Relación señal/ruido extremadamente alta.
- Precisión aumentada de 6 mm.
- Supresión automática de falsos ecos para la eliminación de falsos ecos de componentes fijos.
- Conversión de nivel de relleno/volumen o nivel de relleno/caudal.
- Instalación “*plug and play*”. Conectar y listo.

<sup>1</sup> La hoja de características, así como toda la información relativa a las propiedades eléctricas del sensor pueden ser consultadas en el anexo de hojas de características y manuales.

<sup>2</sup> Podemos entender relleno también como distancia ya que el sensor permite hacer la conversión de distancia a nivel de llenado en un depósito.

### 5.5.2. Instalación y calibración.

Antes de colocar el sensor en el lugar definitivo de medida, debemos conectarlo de forma que esté alimentado eléctricamente y sea capaz de transmitir los datos de medición. Al ser un dispositivo de dos hilos, el cableado es bastante sencillo. En nuestro caso, como ya se ha explicado en el cuarto capítulo de este *PFC*, el sensor está conectado mediante el bus de campo *PROFIBUS PA*. Para conectarlo simplemente tendremos que seguir los siguientes pasos:

- Desenroscar la prensa estopa izquierda de la parte posterior del sensor y los tornillos de la tapa del sensor.
- Pelar el extremo de la cubierta del cable PROFIBUS PA unos 70 mm (2.75"), y pasar los cables a través del prensaestopas.
- Conectar los cables como se indica, atornillar la tapa y sujetar el bus *PROFIBUS PA* con prensaestopas.

La siguiente imagen (Figura 16) muestra los pasos a seguir para un correcto cableado del sensor.

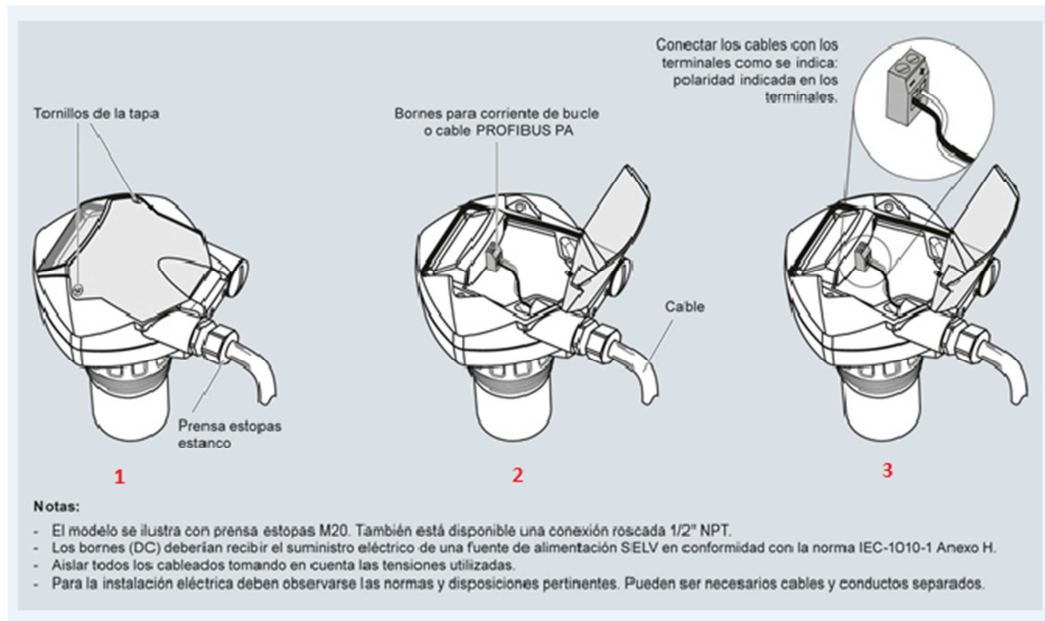


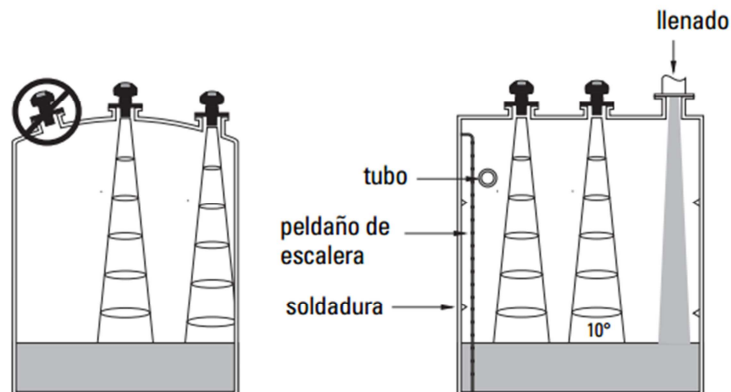
Figura 16: Cableado del sensor SITRANS Probe LU

Una vez cableado el sensor, el siguiente paso es colocarlo en el lugar de trabajo donde debe obtener los datos. Deberemos seguir las siguientes recomendaciones y precauciones (ver *Figura 17*) a la hora de designar el espacio donde va a estar:

- Temperatura ambiente: de -40 a 85 °C.
- El aparato debe situarse en un lugar visible, de fácil acceso (programación con el control remoto).
- Montar el aparato tomando en cuenta el tipo de protección de la carcasa y los materiales de construcción.
- Mantener el trazado sonoro perpendicular a la superficie del material.
- Mantener el trazado sonoro lejos de obstrucciones o aberturas de llenado.
- Se recomienda montar el aparato lejos de los circuitos de alta tensión o corriente, contactores y controladores de velocidad de motor de frecuencia variable.
- Es recomendable instalar el sensor respetando una distancia mínima de 300 mm entre la superficie emisora del sensor y el máximo nivel esperado aunque el rango mínimo sea de 250 mm.

El trazado sonoro debería:

- ser perpendicular a la superficie controlada
- situarse lejos de paredes rugosas, soldaduras, peldaños y otras obstrucciones
- situarse lejos de la corriente de llenado



*Figura 17: Instalación del sensor*

En nuestro caso, como ya se ha explicado en el tercer capítulo, hemos colocado el sensor siguiendo las recomendaciones anteriores sobre una plancheta de acero, sujeta ésta a la maqueta mediante bridas y debidamente brocada para asegurar el dispositivo.

Por otro lado, para calibrar<sup>1</sup> el sensor debemos tener en cuenta las indicaciones que podemos observar en la *Figura 18*:

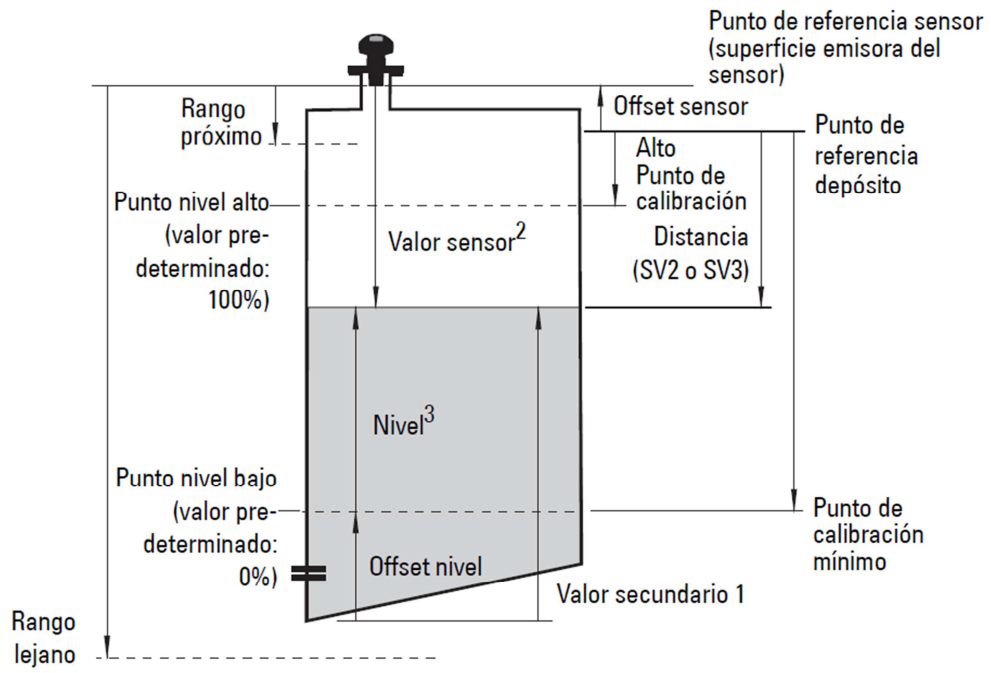


Figura 18: Calibración del sensor

<sup>1</sup> Ver anexo de prácticas de alumnos y manuales del sensor para la calibración del aparato.

Tomando como ejemplo la imagen anterior y las recomendaciones de instalación, el sensor queda acoplado en nuestra maqueta de pruebas tal y como muestra la *Figura 19*.

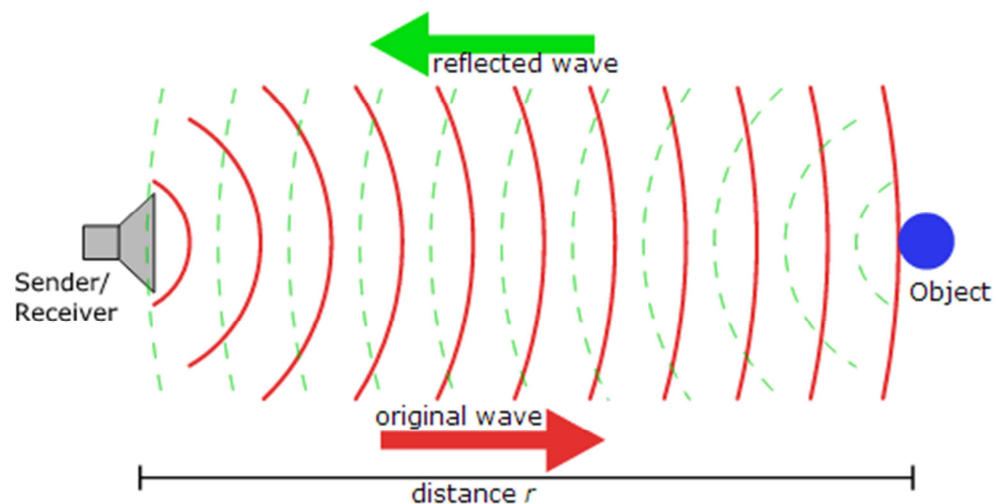


*Figura 19: Sensor instalado y calibrado en la maqueta de pruebas*

### 5.5.3. Funcionamiento.

Para entender el funcionamiento de este tipo de aparatos, hay que tener presente el concepto de radar. Los sensores de ultrasonidos, de hecho, constituyen pequeños radares. Esto se debe gracias a las ondas de ultrasonidos. Dichas ondas al viajar por un medio, en este caso aire, son reflejadas al chocar con cualquier obstáculo o elemento que se encuentre en su camino. Es decir, “rebotan” contra aquello que se encuentre a su paso. Esta reflexión o “rebote” se debe a la diferencia de impedancia acústica entre el medio por el que se propaga la onda y el objeto con el que chocan.

Al emitir un pulso de sonido a una frecuencia tan alta que es inaudible para el oído humano, el sensor calcula el tiempo que tarda el sonido en llegar al obstáculo, “rebotar” y volver de nuevo al emisor. En la siguiente imagen (*Figura 20*) se puede ver como el sensor (emisor) emite las ondas ultrasónicas que se reflejan en un obstáculo a su paso y vuelven de nuevo al dispositivo.



*Figura 20: Emisión y reflexión de las ondas sobre un objeto*

El tiempo transcurrido desde el envío de la onda hasta su recepción, se llama tiempo de eco y se utiliza para calcular la distancia a la que se encuentra el objeto en el que rebota la onda.

Ahora que ya conocemos el funcionamiento básico de un sensor de ultrasonidos, fijémonos en la *Figura 21*:

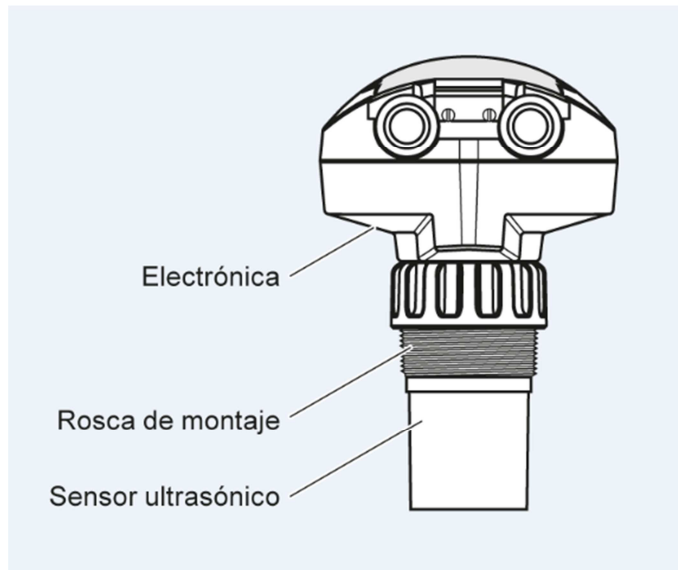


Figura 21: Vista posterior del sensor

No es difícil adivinar donde encontramos la parte ultrasónica encargada de la emisión y recepción de las ondas que a posteriori, servirán para hacer los cálculos necesarios para la medición. Si seccionáramos el emisor-receptor de un sensor de ultrasonidos veríamos la siguiente imagen (Figura 22):

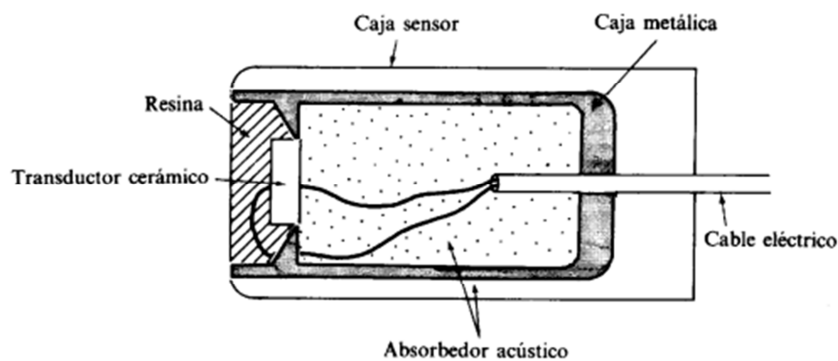


Figura 22: Emisor-receptor seccionado

Como podemos ver, el transductor se encuentra conectado a un cable eléctrico, para así poder emitir los pulsos ultrasónicos. La capa de resina actúa como protector contra el polvo, la humedad y otros factores medioambientales, y el material absorbente se utiliza para amortiguar la energía acústica y detectar objetos cercanos al sensor, ya que el transductor actúa como emisor y receptor al mismo tiempo.

Además, como ya hemos citado antes, una gran ventaja que incorpora el sensor es la supresión automática de falsos ecos, haciendo que el dispositivo sea capaz de reconocer y aprender donde se encuentre cualquier obstáculo distinto al material a medir como pueden ser soldaduras, tuberías... De esta forma también se mejora la relación señal/ruido haciendo al sensor más preciso. En la Figura 23 podemos ver una comparativa entre la señal recibida sin la supresión de falsos ecos y aplicando este método.

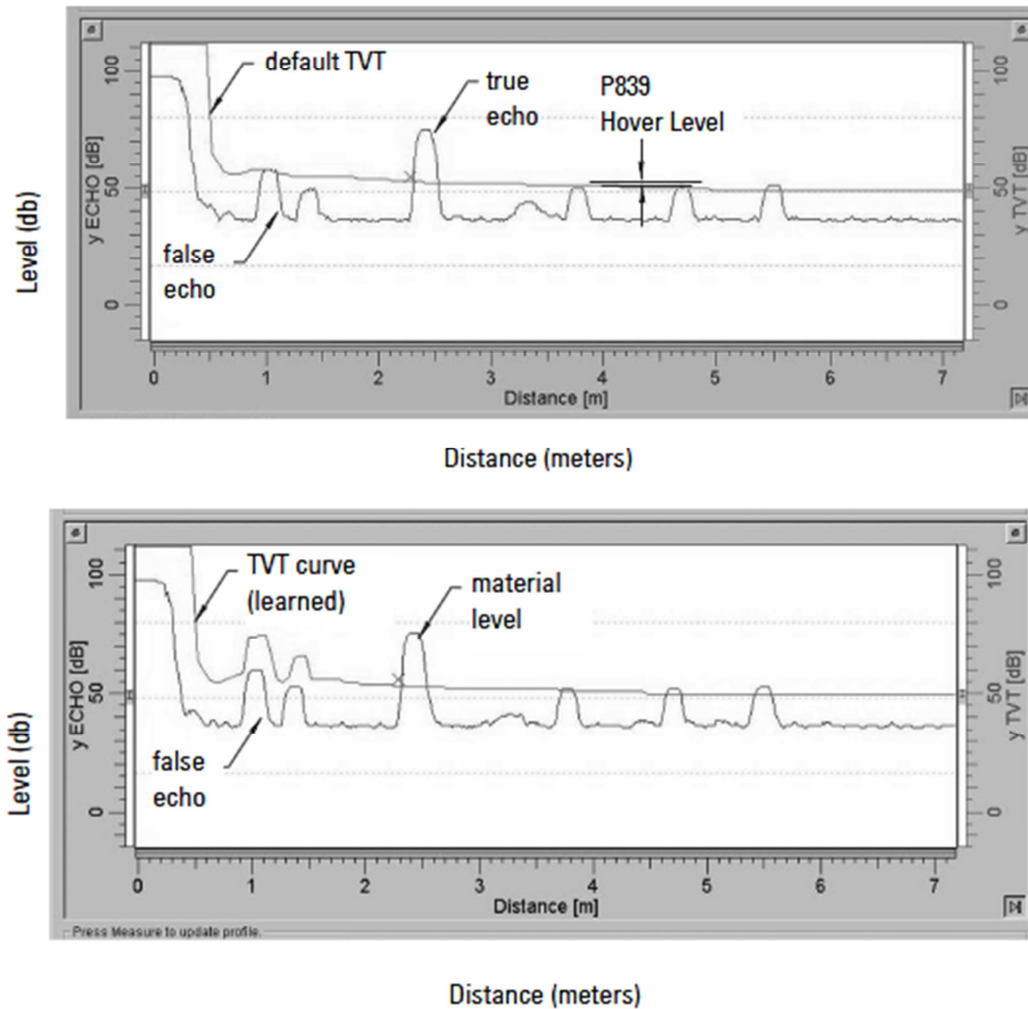


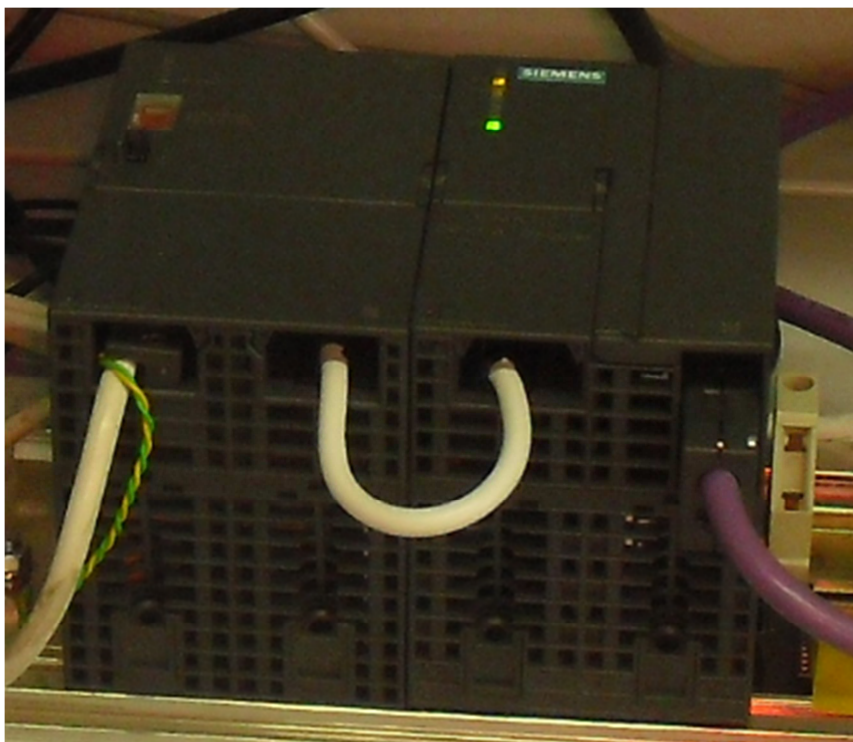
Figura 23: Antes y después de aplicar la supresión de falsos ecos

#### 5.5.4. Comunicación.

Como ya sabemos, el sensor *SITRANS Probe LU* puede comunicarse mediante el protocolo *HART* (ya explicado en este capítulo) o el estándar *PROFIBUS PA*<sup>1</sup>. En nuestro caso, el dispositivo se encuentra comunicado con el sistema de control mediante el bus de campo *PROFIBUS PA*. El sensor utiliza este estándar, ya que está diseñado para trabajar en entornos más agresivos, especialmente en temperaturas elevadas, que el estándar *PROFIBUS DP*.

El resto de componentes que forman el sistema de control se comunican mediante *PROFIBUS DP*, por lo que se hace necesario el uso de una pasarela *PROFIBUS DP/PA* para obtener la información del sensor. En la imagen (*Figura 24*) podemos ver la pasarela utilizada.





*Figura 24: Pasarela PROFIBUS DP/PA*

<sup>1</sup> Consultar el cuarto capítulo de este *PFC* para mayor información del estándar *PROFIBUS PA*.

De esta forma, la estructura hardware de comunicación<sup>1</sup> del sensor quedaría formada por los elementos de control comunicados entre sí con el estándar *DP* de *PROFIBUS*, y el sensor se comunicaría con éstos con el estándar *PA* a través de la pasarela *DP/PA*, tal y como podemos ver en la *Figura 25*.

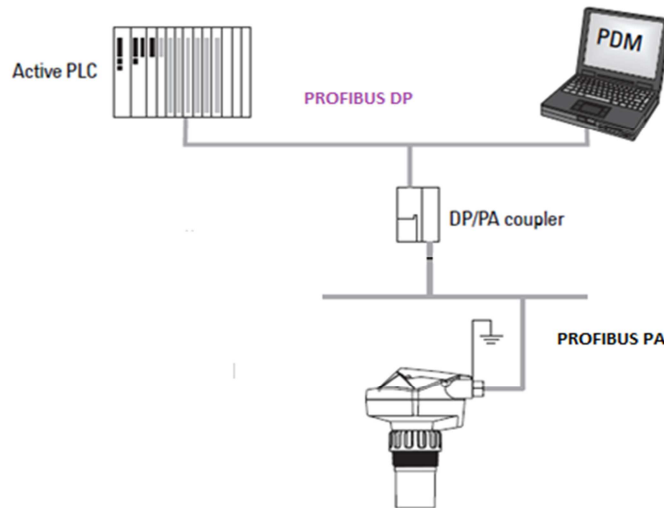
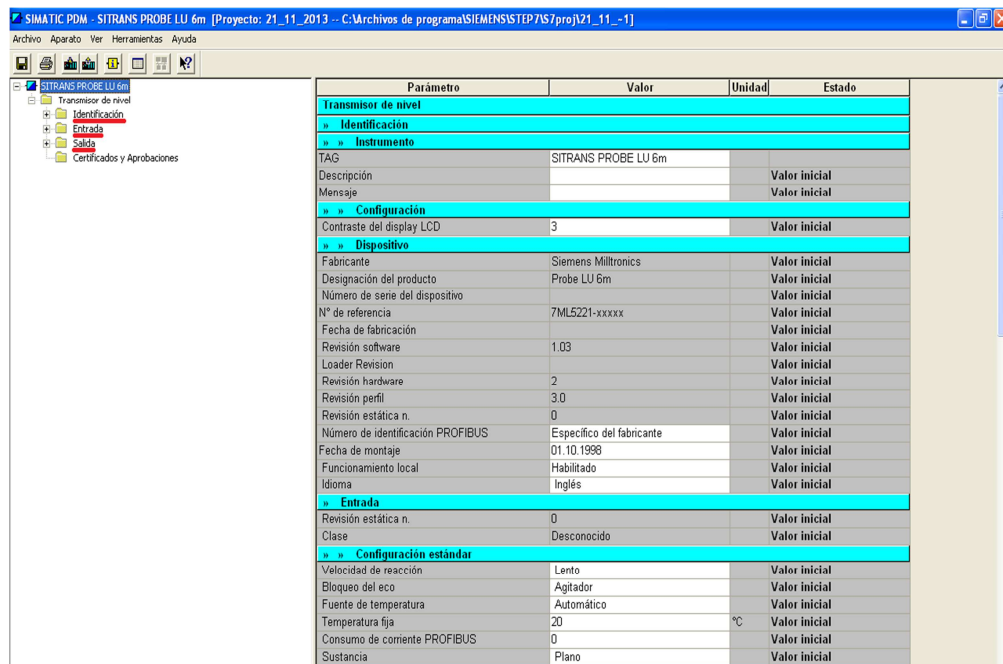


Figura 25: Esquema de las comunicaciones

<sup>1</sup> Consultar los anexos de prácticas de alumnos para saber cómo configurar la comunicación entre el sensor y la estación de control.

### 5.5.5. Software y configuración del sensor.

El software utilizado para configurar el sensor desde la estación de control o *PC* del operador se denomina *SIMATIC PDM*<sup>1</sup> que consta de 3 *CD's*. Con este programa podremos configurar el sensor, calibrándolo y ajustando sus parámetros para obtener la información necesaria y el máximo rendimiento del aparato. Una vez que accedamos a la configuración del sensor, veremos la siguiente imagen (*Figura 26*). Para obtener una información más detallada sobre la configuración del sensor, es necesario consultar los anexos.



Parámetro	Valor	Unidad	Estado
<b>Transmisor de nivel</b>			
<b>Identificación</b>			
<b>Instrumento</b>			
TAG	SITRANS PROBE LU 6m		
Descripción			Valor inicial
Mensaje			Valor inicial
<b>Configuración</b>			
Contraste del display LCD	3		Valor inicial
<b>Dispositivo</b>			
Fabricante	Siemens Miltronics		Valor inicial
Designación del producto	Probe LU 6m		Valor inicial
Número de serie del dispositivo			Valor inicial
Nº de referencia	7ML5221-xxxxxx		Valor inicial
Fecha de fabricación			Valor inicial
Revisión software	1.03		Valor inicial
Loader Revision			Valor inicial
Revisión hardware	2		Valor inicial
Revisión perfil	3.0		Valor inicial
Revisión estática n.	0		Valor inicial
Número de identificación PROFIBUS	Específico del fabricante		Valor inicial
Fecha de montaje	01.10.1998		Valor inicial
Funcionamiento local	Habilitado		Valor inicial
Idioma	Inglés		Valor inicial
<b>Entrada</b>			
Revisión estática n.	0		Valor inicial
Clase	Desconocido		Valor inicial
<b>Configuración estándar</b>			
Velocidad de reacción	Lento		Valor inicial
Bloqueo del eco	Agitador		Valor inicial
Fuente de temperatura	Automático		Valor inicial
Temperatura fija	20	°C	Valor inicial
Consumo de corriente PROFIBUS	0		Valor inicial
Sustancia	Plano		Valor inicial

Figura 26: Interfaz de configuración del sensor

<sup>1</sup> Consultar los anexos de prácticas de alumnos para configurar el sensor usando *SIMATIC PDM*.

# Capítulo 6: Programación del PLC.

El sexto capítulo tratará sobre la programación del *PLC* para el control del sistema físico, así como la realización de una serie de prácticas para el alumnado. Como hemos explicado en el cuarto capítulo de este documento, el software que vamos a utilizar para la programación del autómatas es el software *Step 7* en su versión 5.5 con el service pack 2.

Como ya sabemos, esta herramienta nos permite programar en tres lenguajes distintos; *KOP*, *FUP* y *AWL*. Cada lenguaje tiene sus características particulares y queda para el programador, elegir cuál se adapta más a sus necesidades. Pero podemos afirmar, que cuanto más complejo es el lenguaje de programación, menos código necesitará, y eso se traduce en un ahorro considerable de memoria. En nuestro caso, vamos a programar usando el lenguaje *KOP*, más sencillo e intuitivo para el alumno a la hora de trabajar.

## 6.1. El entorno de Step 7.

Antes de comenzar con la programación en sí, es importante conocer el entorno en el que vamos a trabajar para evitar futuras complicaciones o pérdidas de tiempo buscando los elementos necesarios por desconocer la interfaz de programación. Una vez que accedemos al programa (ver anexos de prácticas) veremos la siguiente interfaz (*Figura 1*):

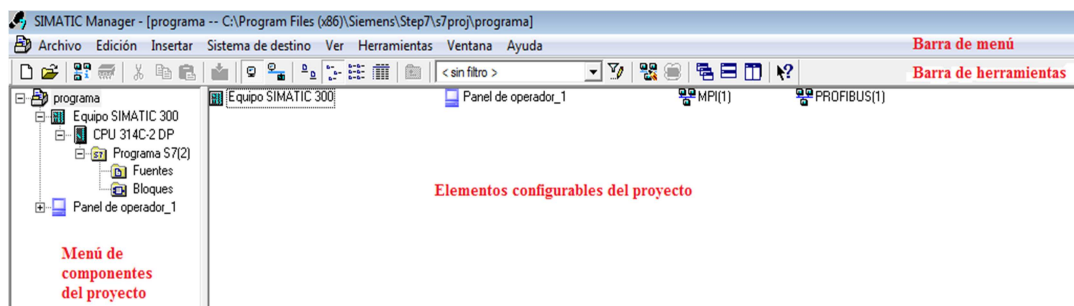


Figura 1: Interfaz de trabajo

Como observamos en la imagen anterior, la interfaz se puede dividir en varias zonas donde encontramos los elementos necesarios para nuestro trabajo. Estas divisiones son:

- Barra de menú: aquí podremos realizar todas las acciones relativas a nuestro trabajo, tales como acceder a la tabla de símbolos, cargar la configuración al *PLC*, etc...
- Barra de herramientas: proporciona un acceso directo a las opciones destacadas de la barra de menú.
- Menú de componentes del proyecto: desde aquí podremos editar o añadir más componentes a nuestro trabajo.
- Elementos configurables del proyecto: a través de ellos accederemos a las distintas interfaces de configuración de nuestro trabajo.

De esta forma, ya tendremos una idea global de la interfaz y de dónde podemos encontrar los elementos o emprender las acciones necesarias para la realización del proyecto.

### 6.1.1. Interfaz de configuración hardware.

Esta interfaz es una parte importante de cada proyecto a realizar. Aquí configuramos el hardware que intervendrá en nuestro trabajo. Es decir, recreamos de forma virtual, todo los componentes físicos que forman parte de nuestro proyecto. Es sumamente importante no cometer fallos a la hora de realizar la configuración hardware, ya que cargaremos en el autómatas los elementos que formen dicha configuración, por lo que basta decir que han de corresponderse con los elementos físicos.

Para acceder a la interfaz, en primer lugar iremos al *menú de componentes del proyecto* de la interfaz de inicio. Seguidamente pincharemos sobre *Equipo SIMATIC 300* y por último haremos doble clic sobre *Hardware* como observamos en la imagen a continuación (Figura 2):

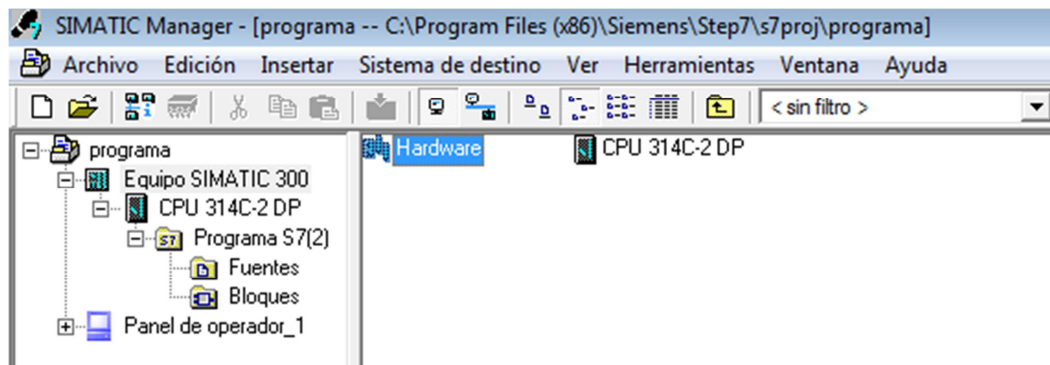


Figura 2: Acceder a la interfaz de configuración hardware

Una vez que abramos la interfaz de configuración hardware, aparecerá la siguiente ventana (Figura 3):

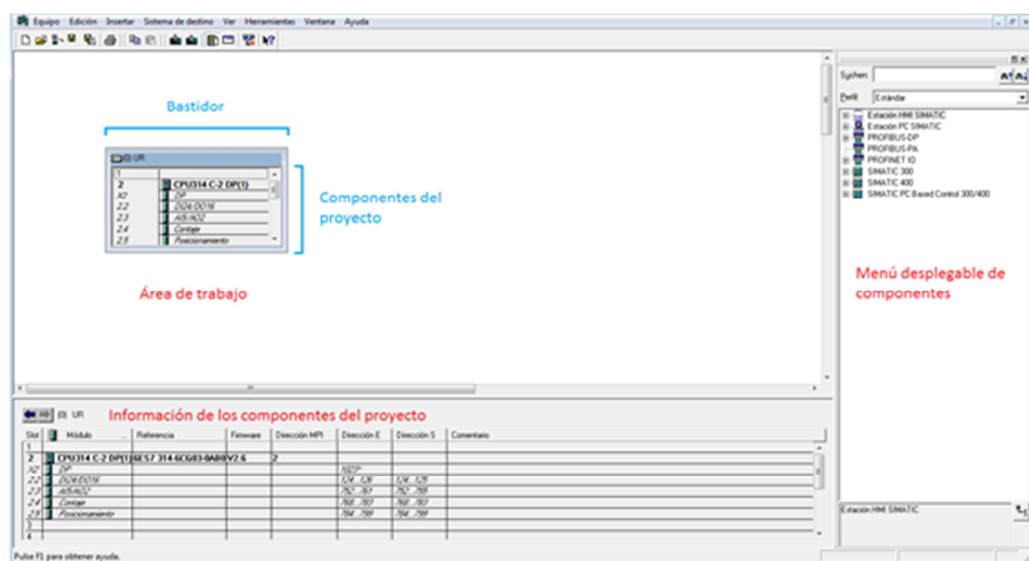


Figura 3: Interfaz de configuración hardware

Al observar la imagen anterior, vemos que la interfaz se puede dividir en tres partes claramente diferenciadas:

- Área de trabajo: donde se configura la arquitectura hardware del proyecto.
- Información de los componentes del proyecto: cantidad de *E/S*, direcciones de las mismas, etc.
- Menú desplegable de componentes: donde aparecen los distintos componentes con los que configurar nuestra arquitectura hardware.

### 6.1.2. Interfaz de programación (OB1).

La última de las interfaces importantes para empezar a trabajar con *Step 7* es la Interfaz de programación. Aquí configuraremos el programa o código que será cargado en el autómatas para controlar nuestro sistema físico. Para poder acceder a la interfaz, debemos abrir el bloque de organización (*OB1*) que será realmente donde programaremos. Desplegaremos el *menú de componentes del proyecto* como se observa en la siguiente imagen, y a continuación haremos doble clic sobre el *OB1* tal y como muestra la *Figura 4*.

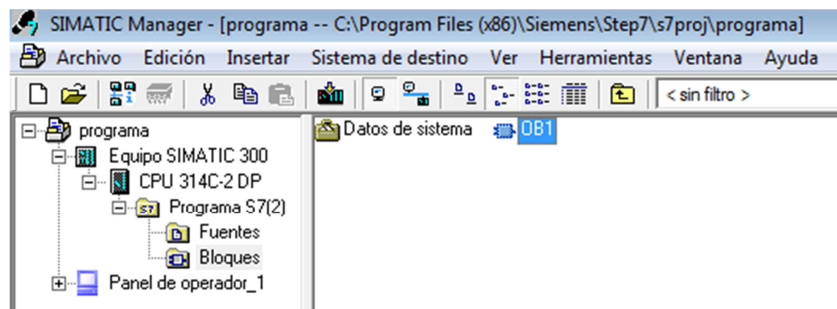


Figura 4: Acceder a la interfaz de programación (OB1)

Al abrir el bloque de organización (*OB1*) aparecerá la siguiente interfaz (*Figura 5*) donde realizaremos el programa de control.

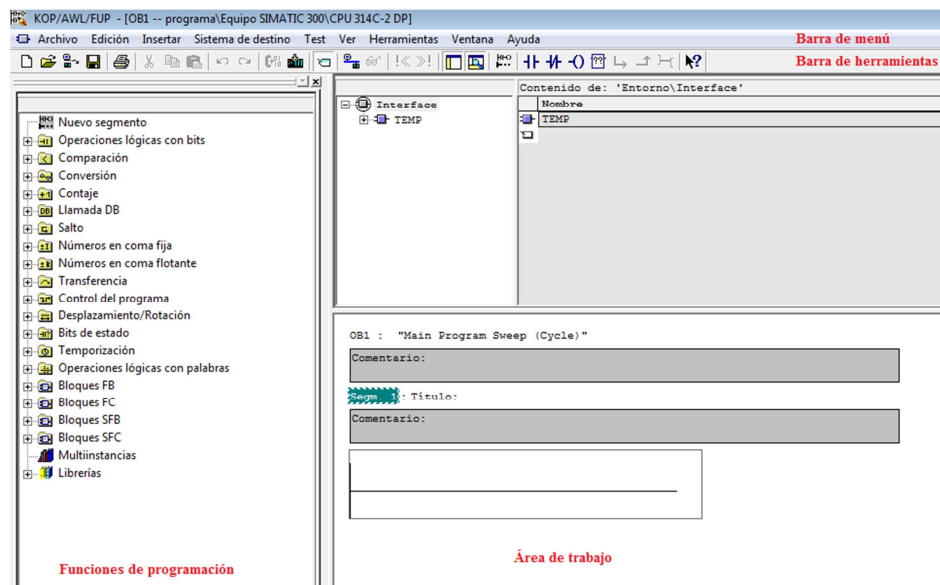







Figura 5: Interfaz de programación

Dentro de la interfaz de programación podemos destacar tres zonas, diferenciadas entre sí al observar la imagen anterior:

- La barra de herramientas: donde se encuentran algunas herramientas necesarias para la programación.
- Las funciones de programación: donde se encuentran algunos de los elementos que vamos a utilizar en la programación.
- El área de trabajo: donde programaremos todo el código necesario.

### Conceptos básicos.

-  Entrada digital activa.
-  Entrada digital a cero.
-  Salida digital activa.
-  Insertar segmento nuevo.
-  Cargar en el autómata.

Para obtener más información del entorno de Step 7, se recomienda consultar los anexos de prácticas y manuales de dicho software.

## 6.2. Configuración hardware.

Antes de empezar a programar, es necesario realizar la configuración hardware del sistema. Para ello, debemos acceder a la interfaz de configuración hardware (*Figura 2 y 3*) y una vez ahí configurar la arquitectura hardware.

Si nos fijamos en la ventana de *Información de los componentes*, en el segundo espacio, aparece la referencia del autómata. Esta referencia nos va a indicar, entre otras cosas, los módulos acoplados (por ejemplo: *E/S* digitales o analógicas) a la *CPU* del aparato. Esta referencia no coincide con la referencia del autómata que tenemos en el laboratorio. Recordemos que *Step 7*, en su configuración Hardware, por defecto nos asigna una *CPU*. Pero la referencia de ésta no tiene porqué ser la que tenemos en nuestro puesto de trabajo. Por lo tanto, tendremos que sustituirla por la adecuada.

Antes de nada, debemos eliminar la *CPU* errónea. En el **Área de trabajo** pinchamos sobre el bastidor y seleccionamos la *CPU* que se encuentra remarcada en negrita. A continuación hacemos clic sobre ella con el segundo botón del ratón y seleccionamos la opción **Borrar** del menú desplegable. Confirmamos la elección siguiendo los pasos como en la imagen (*Figura 5*):

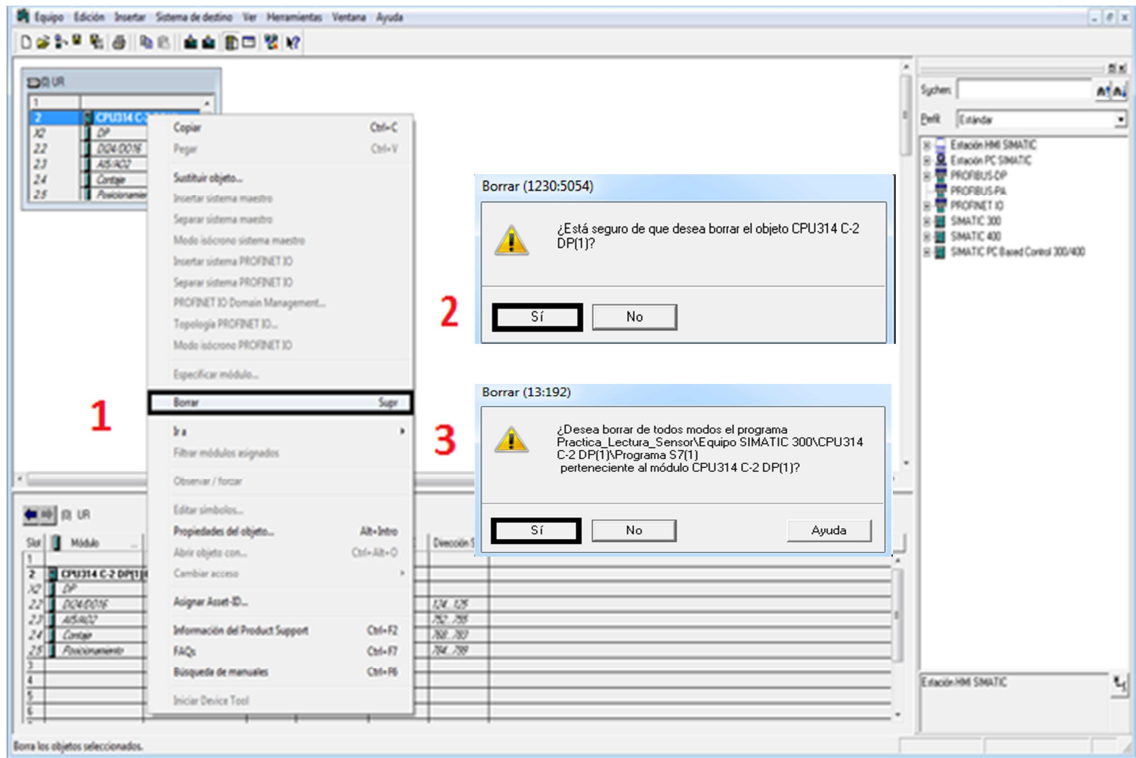


Figura 5: Eliminar CPU

También se puede eliminar la *CPU* seleccionándola en el bastidor y apretando la tecla **Supr** (suprimir) del teclado.

Una vez hayamos eliminado el autómata incorrecto, sólo tendremos el bastidor en el que colocar los elementos necesarios. Ya podemos incluir la *CPU* deseada. Para ello vamos a la ventana donde está el *Menú desplegable de componentes* y desplegamos el apartado **SIMATIC 300**. A continuación abrimos el menú **CPU-300** y clicamos en **CPU-314C-2 DP**. Al desplegarse, seleccionamos el primer componente que aparece, cuya referencia es **6ES7 314-6CF00-0AB0**. Nada más pinchar en ella el programa nos hará saber dónde debemos colocarla resaltando su lugar en color verde. Pinchamos y arrastramos hasta el lugar indicado o hacemos doble clic.

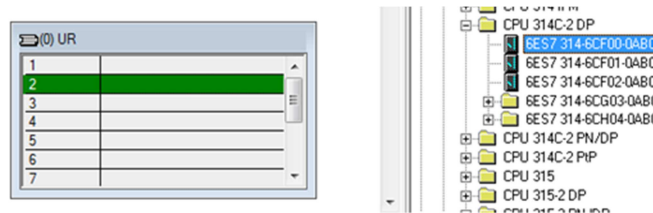


Figura 6: Detalle del bastidor y el lugar de la CPU



Cuando hagamos esto, el programa, automáticamente nos preguntará a qué red queremos conectar nuestro autómatas:

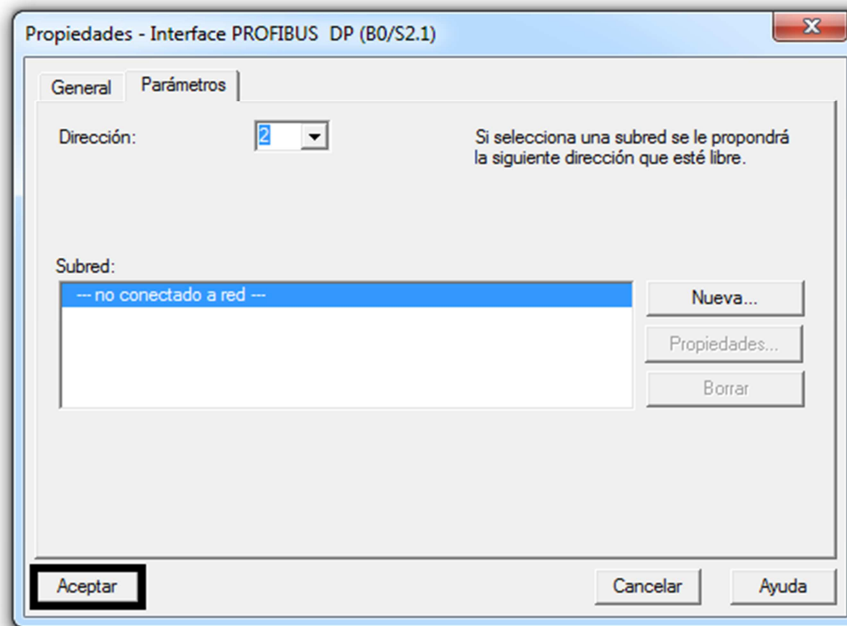


Figura 7: Conectar el autómatas a una red

Simplemente aceptaremos y continuaremos con la configuración. Más adelante se explicará cómo añadir y configurar una red. El programa nos permite integrar una red a un *PLC* cuando configuramos un autómatas nuevo, o, a partir de un *PLC* ya existente en nuestra configuración Hardware, acoplarle una red. En ambos casos, la configuración de dicha red se hace de manera idéntica.

El siguiente paso será insertar en el bastidor la fuente de alimentación del autómatas. Para ello, vamos al menú desplegable de componentes y abrimos el menú **SIMATIC 300**. A continuación desplegamos el menú **PS – 300** y elegimos la fuente deseada; **PS 307 5** como vemos en la Figura 8.

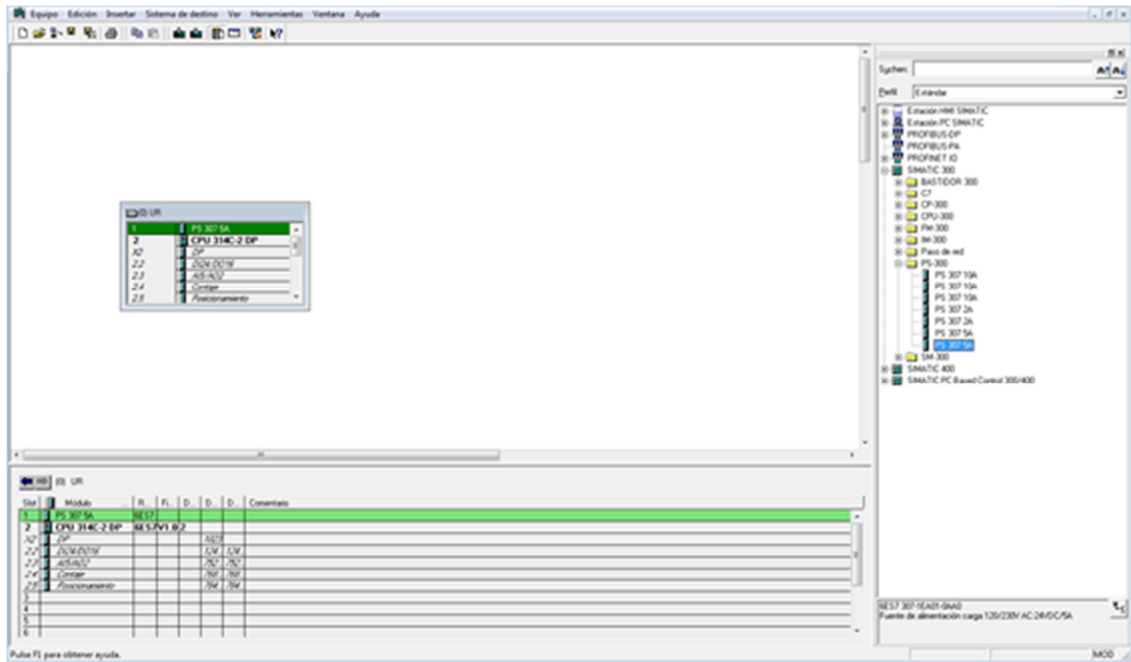


Figura 8: Añadir la fuente de alimentación

Ahora ya podemos conectar una red de comunicación a nuestro *PLC*. A dicha red también conectaremos el sensor inteligente, pero eso lo veremos en el siguiente paso. Para configurar una red, simplemente tenemos que ir al bastidor. Si nos fijamos bien, en el segundo espacio se encuentra la *CPU* que hemos insertado con anterioridad. Dicho espacio se divide en “subespacios” en los que podemos ver los módulos de Entradas/Salidas tanto digitales como analógicas, contadores, etc. integrados con el autómata. Cada “subespacio” corresponde a un módulo físico integrado en la realidad.

La red que vamos a conectar estará formada por un bus de campo regido por el estándar de comunicación **PROFIBUS DP**. Por lo tanto, para insertar dicho bus, iremos al **bastidor**, y en el “subespacio” X2 destinado a las comunicaciones, pincharemos con el segundo botón en la casilla **DP**. En el menú desplegable elegiremos **Insertar sistema maestro** (Figura 9) apareciendo a continuación la ventana de propiedades de la red (Figura 10).

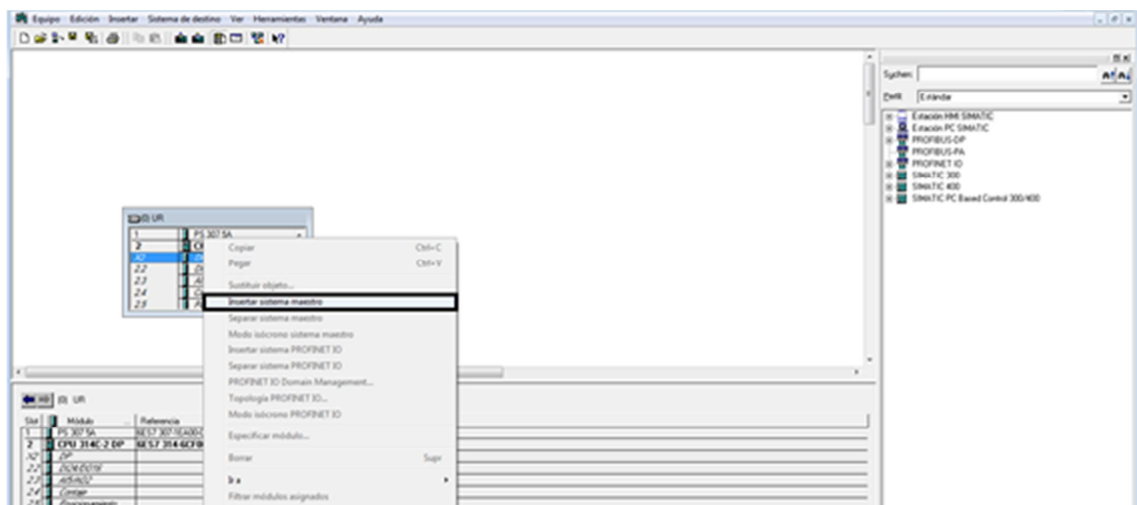


Figura 9: Insertar una red de comunicación

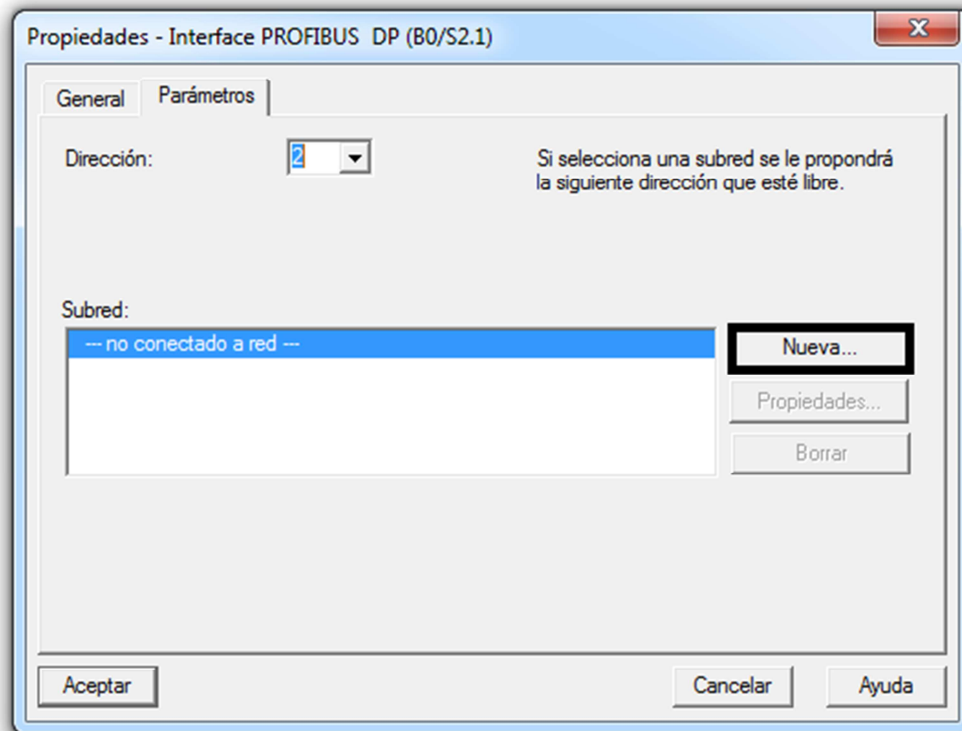


Figura 10: Propiedades de la red

En la ventana de propiedades de la red podemos ver dos pestañas. En la pestaña *General* podremos acceder a una breve información sobre la red así como añadir algún comentario si lo deseamos, mientras que en la pestaña *Parámetros* (Figura 10) aparecerán las redes que tengamos configuradas con anterioridad, en caso de tenerlas, así como la dirección del autómatas en dicha red. También tenemos la posibilidad de crear una conexión nueva, y precisamente esto es lo que vamos a hacer. Pincharemos en el botón **Nueva...** para configurar una nueva red a la que conectar el PLC.

Al pinchar en **Nueva...** aparecerá el siguiente cuadro donde como podemos ver, también aparecen dos pestañas. En la pestaña *General* podremos cambiar el nombre de la red y añadir algún comentario. Seleccionaremos la pestaña *Ajustes de la red* y estableceremos la velocidad de transferencia a **45,45 (31,25) kbit/s**. Colocamos la Velocidad de transferencia a 45,45 kbits/s porque es la velocidad máxima de transferencia de *PROFIBUS PA*. Recordemos que nos comunicamos con el sensor *PROBE LU* mediante una pasarela *PROFIBUS DP/PA*. Además nos aseguraremos de que el perfil indicado es **DP**. Pinchamos en **Aceptar** y el programa nos preguntará si queremos conectar el autómatas a la red que acabamos de configurar (Figura 11). Pincharemos en **Aceptar** para confirmar la orden.

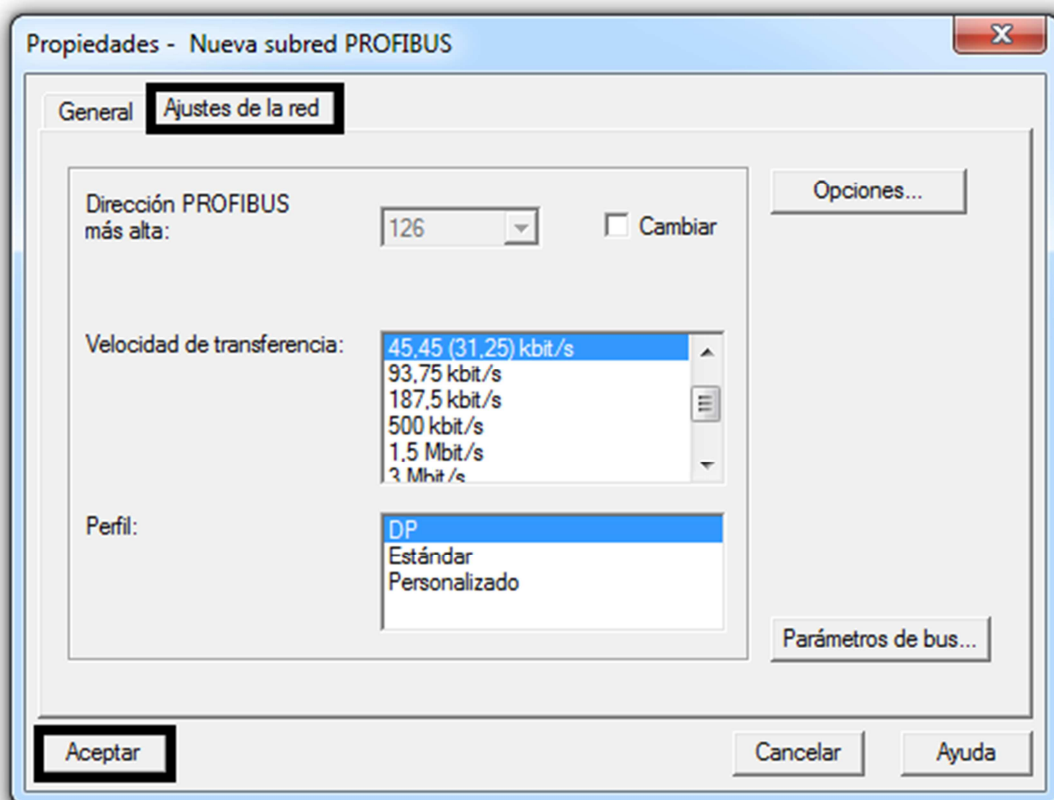


Figura 10: Editar configuración

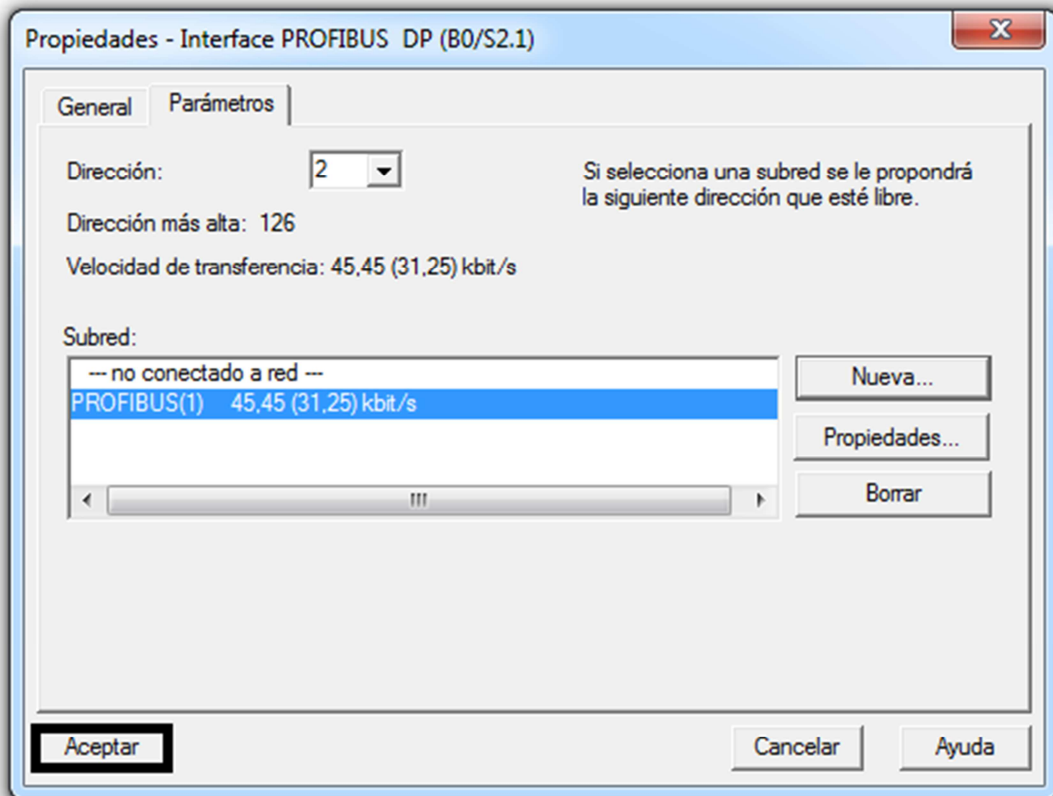


Figura 11: Conectar el PLC a la red creada

De esta manera habremos creado, configurado e insertado una red de comunicación en la configuración Hardware de nuestro proyecto como así lo verifica la imagen siguiente (Figura 12):

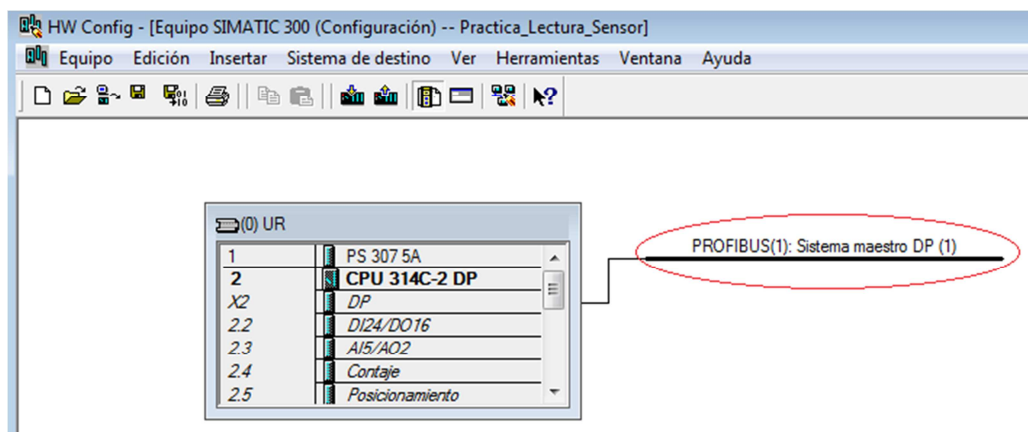


Figura 12: Detalle de la red PROFIBUS DP conectada al autómata

Ahora que ya hemos implementado una red *PROFIBUS*, el siguiente paso y último para completar nuestra configuración Hardware, será insertar el sensor inteligente *Sitrans PROBE LU* a dicha red. Para ello vamos al *Menú desplegable de componentes*, en la parte derecha de la configuración Hardware, y hacemos doble clic en **PROFIBUS PA**. A continuación procederemos de la misma forma siguiendo el esquema: *Sensores* → *Nivel* → *Eco* → *SIEMENS Milltronics* hasta llegar al dispositivo *SITRANS PROBE LU 6m*.

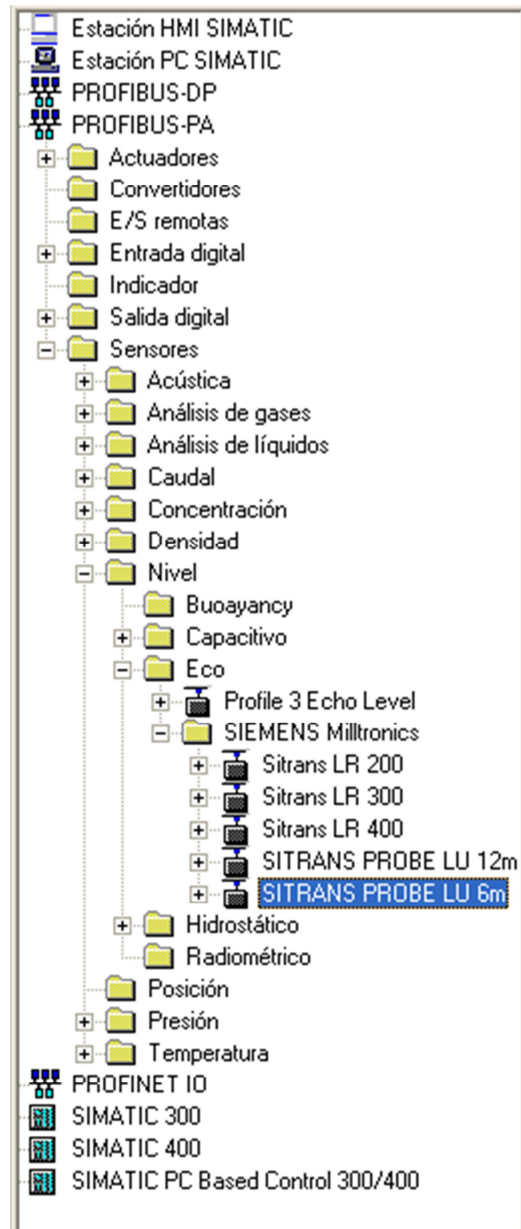


Figura 13: Detalle acceso al sensor

Para conectar el sensor a la red, simplemente tenemos que pinchar sobre él y sin soltar el puntero, arrastrarlo hasta la conexión. Cuando soltemos el puntero para conectarlo a la red se abrirá la ventana de propiedades de conexión del sensor:

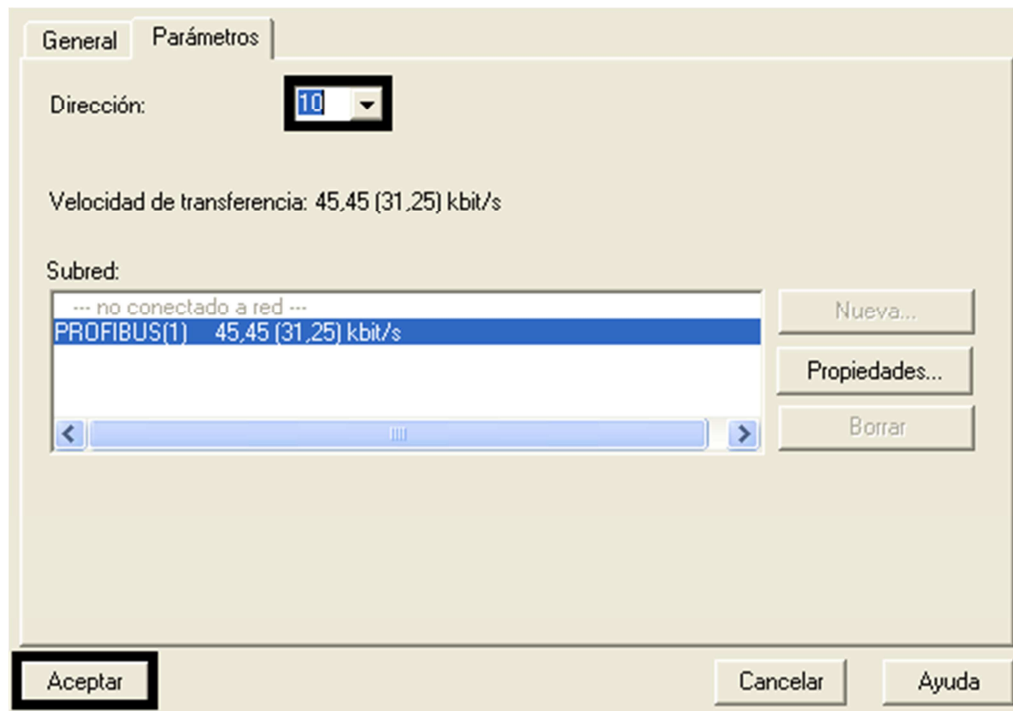


Figura 14: Propiedades de conexión del sensor

Como puede verse, también aparecen dos pestañas. Seleccionaremos la pestaña de *Parámetros* y ajustaremos la dirección del sensor al valor 10. Elegimos esta dirección porque es el valor que se ajustó al sensor con el control remoto que incorpora el mismo. Luego seleccionaremos la conexión que creamos (Figura 21) para indicar al programa que conecte al sensor a esa red y pinchamos en **Aceptar** para confirmar.

De esta forma ya tenemos incorporado a la red el sensor *SITRANS Probe LU* como se ve en la imagen (Figura 15):

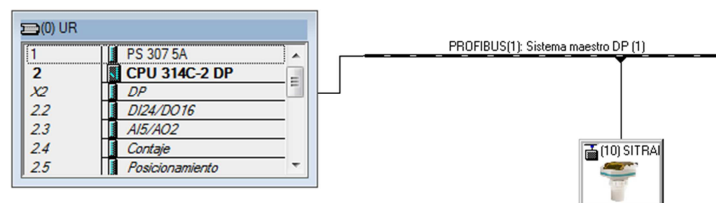



Figura 15: Detalle del sensor conectado a la red

Para acabar, guardaremos la configuración y la cargaremos en el *PLC* pinchando sobre  (*Cargar en módulo*). Cuando carguemos la configuración el programa nos preguntará donde cargar la configuración:

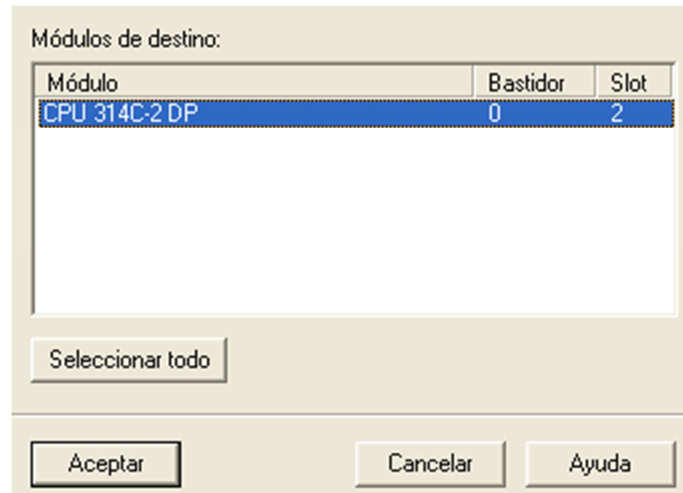
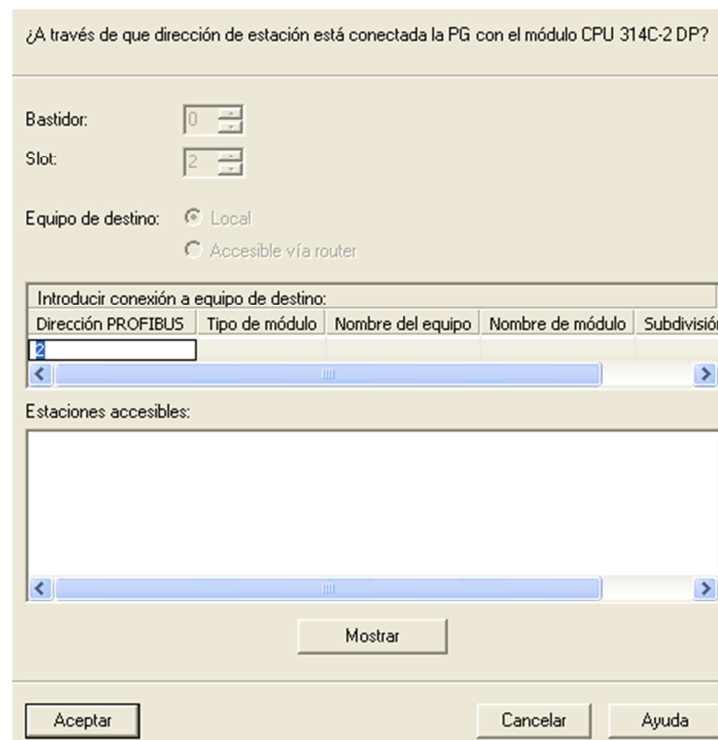


Figura 16: Módulo de destino

Pincharemos en **Aceptar** y en la siguiente ventana que aparezca, haremos clic en **Mostrar** para ver los dispositivos activos en nuestra conexión. Después seleccionaremos el autómata y volveremos a pinchar en **Aceptar**.





¿A través de que dirección de estación está conectada la PG con el módulo CPU 314C-2 DP?

Bastidor:

Slot:

Equipo de destino: ☒ Local  
☐ Accesible vía router

Introducir conexión a equipo de destino:

Dirección PROFIBUS	Tipo de módulo	Nombre del equipo	Nombre de módulo	Subdivisión
2	CPU 314C-2...			

Estaciones accesibles:

Dirección PROFIBUS	Tipo de módulo	Nombre del equipo	Nombre de módulo	Subdivisión
2	CPU 314C-2...			
10	SITRANS P...			

Actualizar

Aceptar Cancelar Ayuda

Figura 17: Cargar en el autómeta

A continuación, el programa necesitará parar el *PLC*. Aceptaremos y seguiremos con la carga:

Para cargar los datos de sistema se paran los siguientes módulos.

Módulo	Bastidor	Slot
CPU 314C-2 DP	0	2

Aceptar Cancelar Ayuda

Figura 18: Parar módulo de destino

El último paso será arrancar de nuevo el autómeta. Confirmaremos la orden pinchando en *Sí* como muestra la *Figura 19*:

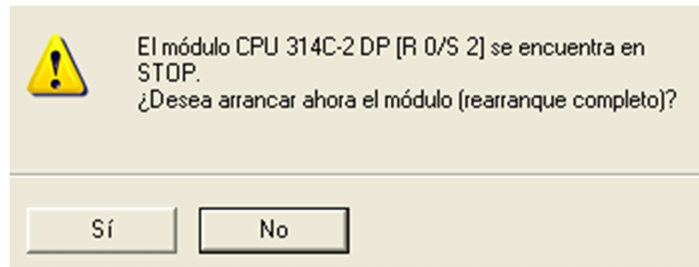


Figura 19: Arrancar el autómeta

De esta forma ya habremos creado y cargado nuestra configuración Hardware. Por último, cerraremos el asistente de configuración.

### 6.3. Ajustar la interfaz PG/PC.

Por último, vamos a comprobar la interfaz *PG/PC* para saber si hemos hecho bien la configuración hardware. La interfaz mostrará los elementos activos dentro del bus de comunicación.

Para abrir la interfaz pincharemos sobre **Inicio** en la barra de herramientas del escritorio y seleccionaremos **Ajustar interface PG/PC**.



Figura 20: Interfaz PG-PC

De esta forma accedemos a la interfaz. Lo siguiente que haremos será comprobar si está bien ajustada (Figura 21):

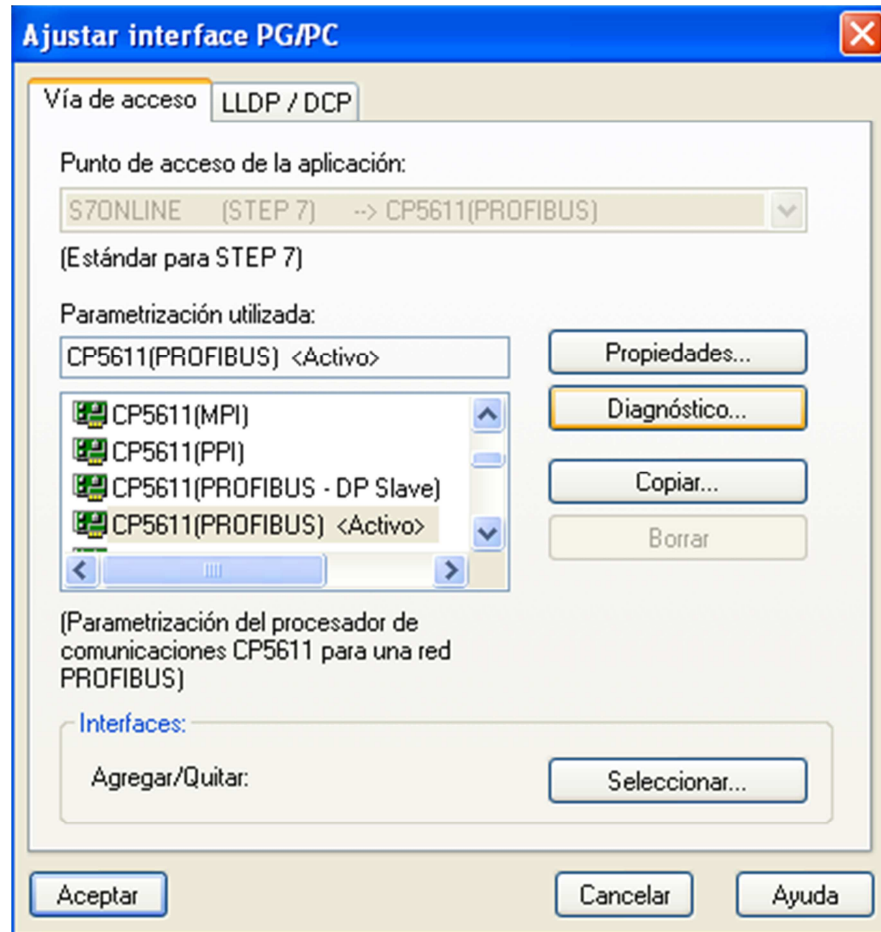


Figura 21: Ajustar la interfaz

Pincharemos en **Propiedades** y ajustaremos la interfaz según muestra la siguiente imagen, en caso de ser necesario:

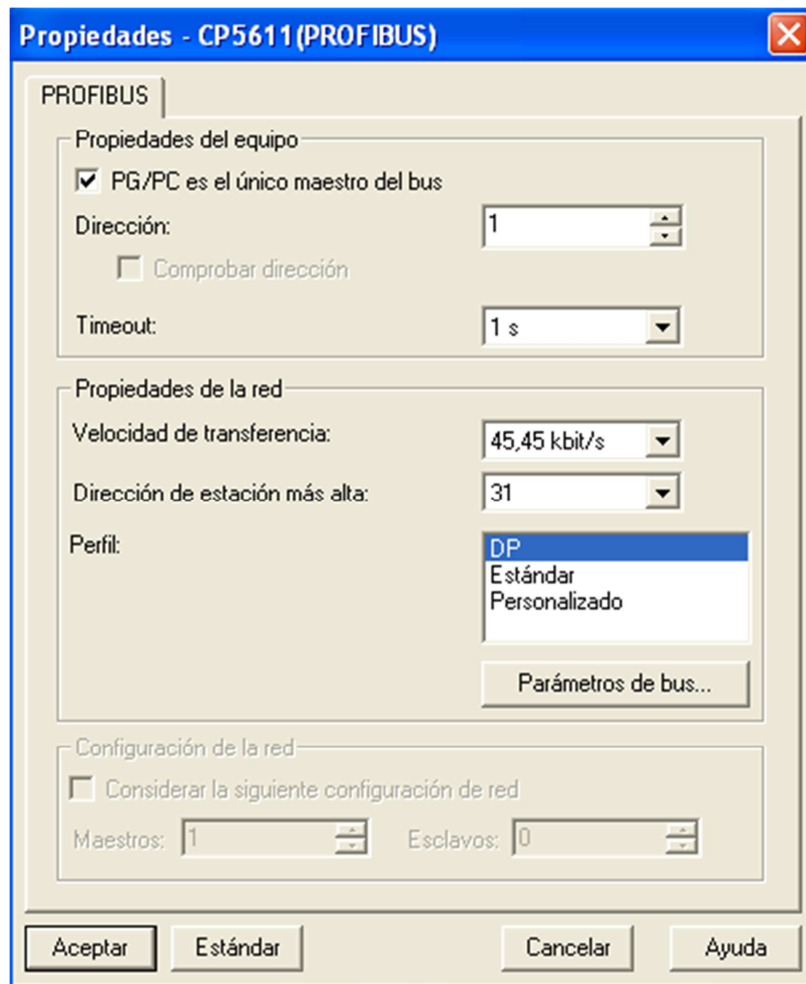


Figura 22: Propiedades de la interfaz

Como ya se ha comentado en la configuración Hardware, ajustamos la **velocidad de transferencia** a **45,45 kbits/s** ya que esta es la velocidad de transferencia de **PROFIBUS PA**. También debe estar marcada la interfaz **PG/PC** como único maestro del bus. Por defecto, la dirección asignada a la interfaz es la número 1. La dejaremos tal y como está. Cuando hayamos hecho las comprobaciones pertinentes, pincharemos en **Aceptar**.

Después de ajustar las propiedades, pincharemos en **Diagnóstico** (Figura 21) y nos aseguraremos de que la interfaz detecta todos los componentes de nuestra configuración Hardware. Para ello pincharemos en **Read** en la ventana que aparece:

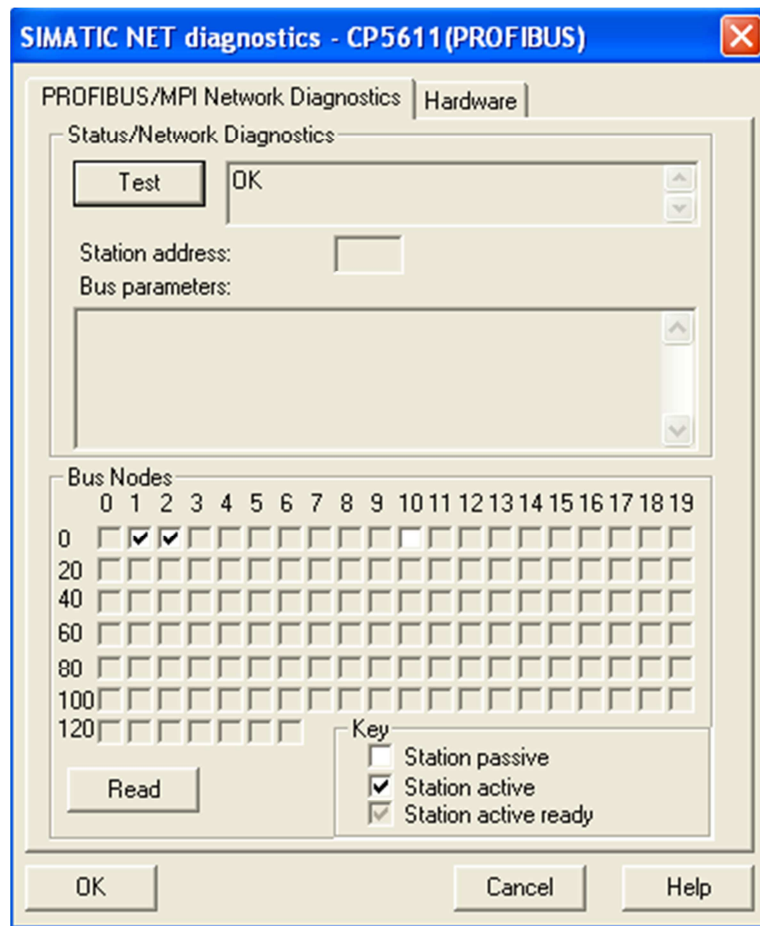


Figura 23: Diagnóstico de la interfaz

Si no hemos cometido fallos, la interfaz mostrará como estaciones activas a nuestro PC, donde se encuentra la tarjeta de comunicación (dirección 1), y al autómata (dirección 2). Y como estación pasiva, a nuestro sensor (dirección 10). Pincharemos en **OK** y cerraremos la interfaz.

#### 6.4. Asociación de variables.

En este punto vamos a configurar la tabla de símbolos necesaria para toda la programación que vamos a realizar. A cada elemento (sensor, actuador...) le asignaremos una dirección de memoria del autómata que se corresponderá con una salida o entrada en el módulo de entradas y salidas digitales o analógicas. La tabla se muestra en la *Figura 24*:

Programa S7(1) (Símbolos) -- prueba6\_10\_2014\Equipo SIMATIC 300\CPU314 C-2 DP(1)

	Estado	Símbolo /	Dirección	Tipo de dato	Comentario
1		Cycle Execution	OB 1	OB 1	
2		DPD_DAT	SFC 14	SFC 14	Read Consistent Data of a Standard DP Slave
3		AoM	E 124.6	BOOL	Modo automático o manual.
4		BOMBA	A 124.0	BOOL	Bomba hidráulica del sistema.
5		EV. Inf. T1	A 124.2	BOOL	Electroválvula inferior del tanque 1.
6		EV. Inf. T2	A 124.4	BOOL	Electroválvula inferior del tanque 2.
7		EV. Sup. T1	A 124.1	BOOL	Electroválvula superior del tanque 1.
8		EV. Sup. T2	A 124.3	BOOL	Electroválvula superior del tanque 2.
9		Flot. T1 N. Máx.	E 124.0	BOOL	Indicador de nivel máximo del tanque 1.
10		Flot. T2 N. Máx.	E 124.1	BOOL	Indicador de nivel máximo del tanque 2.
11		Flot. T3 N. Máx.	E 124.4	BOOL	Indicador de nivel máximo del tanque 3.
12		Marca AoM	M 30.6	BOOL	Marca Modo automático o manual.
13		Marca BOMBA	M 31.0	BOOL	Marca Bomba hidráulica del sistema.
14		Marca EV. Inf. T1	M 31.2	BOOL	Marca Electroválvula inferior del tanque 1.
15		Marca EV. Inf. T2	M 31.4	BOOL	Marca Electroválvula inferior del tanque 2.
16		Marca EV. Sup. T1	M 31.1	BOOL	Marca Electroválvula superior del tanque 1.
17		Marca EV. Sup. T2	M 31.3	BOOL	Marca Electroválvula superior del tanque 2.
18		Marca Flot. T1 N. Máx.	M 30.0	BOOL	Marca Indicador de nivel máximo del tanque 1.
19		Marca Flot. T2 N. Máx.	M 30.1	BOOL	Marca Indicador de nivel máximo del tanque 2.
20		Marca Flot. T3 N. Máx.	M 30.4	BOOL	Marca Indicador de nivel máximo del tanque 3.
21		Marca Opt. T1 N. Mín.	M 30.2	BOOL	Marca Indicador de nivel mínimo del tanque 1.
22		Marca Opt. T2 N. Mín.	M 30.3	BOOL	Marca Indicador de nivel mínimo del tanque 2.
23		Marca R. Cal.	M 31.5	BOOL	Marca Resistencia calorífica.
24		Marca S/F	M 30.5	BOOL	Marca Sin función (futurible óptico del tanque3).
25		Marca Serpentin	M 31.6	BOOL	Marca Máquina de frío.
26		Marca STOP	M 30.7	BOOL	Marca Parada de emergencia.
27		Opt. T1 N. Mín.	E 124.2	BOOL	Indicador de nivel mínimo del tanque 1.
28		Opt. T2 N. Mín.	F 124.3	BOOL	Indicador de nivel mínimo del tanque 2.

Figura 24: Tabla de símbolos

Entradas Digitales	Dirección
Flot. T1 N. Máx.	E 124.0
Flot. T2 N. Máx.	E 124.1
Opt. T1 N. Mín.	E 124.2
Opt. T2 N. Mín.	E 124.3
Flot. T3 N. Máx.	E 124.4
S/F	E 124.5
AoM	E 124.6
STOP	E 124.7

Salidas Digitales	Dirección
BOMBA	A 124.0
EV. Sup. T1	A 124.1
EV. Inf. T1	A 124.2
EV. Sup. T2	A 124.3
EV. Inf. T2	A 124.4
R. Cal.	A 124.5
Serpentin	A 124.6

Entradas Analógicas	Dirección
Sonda de nivel	PEW 752
Pt100 T1	PEW 754
Pt100 T2	PEW 756
Pt100 T3	PEW 758

Salidas Analógicas	Dirección
--------------------	-----------

Válvula motorizada	PAW 752
Válvula motorizada*	PAW 754

\*Al disponer de una única salida analógica, podemos usar cualquiera de las dos salidas que nos ofrece el entrenador.

## 6.5. Aplicaciones básicas. Comprobar E/S y lectura del sensor.

El siguiente paso una vez creada nuestra tabla de variables, será desarrollar un programa muy sencillo para comprobar las E/S que hemos configurado anteriormente y obtener la lectura del sensor inteligente (*Figura 25*).

OB1 : "Main Program Sweep (Cycle)"

Programa sencillo para comprobar el funcionamiento básico de la maqueta.

**Segm. 1:** electroválvula superior tanque 1

Activamos el flotador del tanque 3. De esta forma abrimos la electro-válvula superior del tanque 1 y cerramos su válvula inferior.

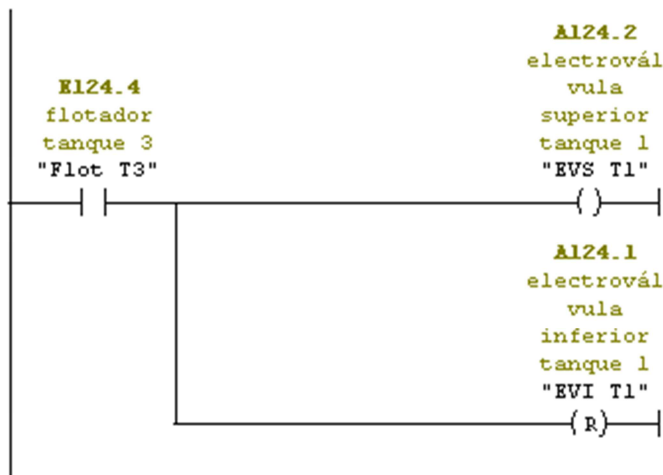


Figura 25: a) Flotador del tanque 3 y electroválvulas del tanque 1

**Segm. 2:** Arrancar motor

Al abrir la electro-válvula superior del tanque 1, arrancamos la bomba. Así evitamos meter presión a las tuberías.



Figura 25: b) Electroválvula superior del tanque 1 y bomba

**Segm. 3:** Parar motor

Al activarse el flotador del tanque 1 (nivel máximo), se parará la bomba y se abrirá la electroválvula inferior del mismo tanque.

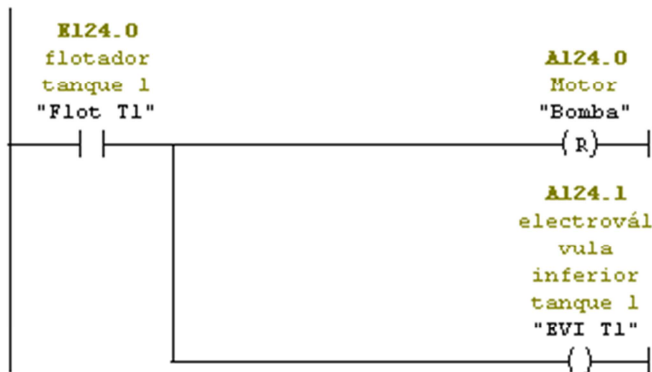


Figura 25: c) Flotador y electroválvula inferior del tanque 1 y bomba

Después de comprobar las E/S vamos a tomar lectura de las mediciones del sensor desde la interfaz de programación. Para poder acceder a las mediciones del sensor necesitamos la ayuda de la función **SFC 14**. Esta función es específica para la lectura de aparatos pasivos en una red Maestro-Esclavo de PROFIBUS DP.

Para insertar la función debemos ir a las funciones de programación, en la parte izquierda de nuestra interfaz y hacer doble clic en el apartado **Librerías**. Luego seguiremos el esquema **Standard Library** → **System Function Blocks**.

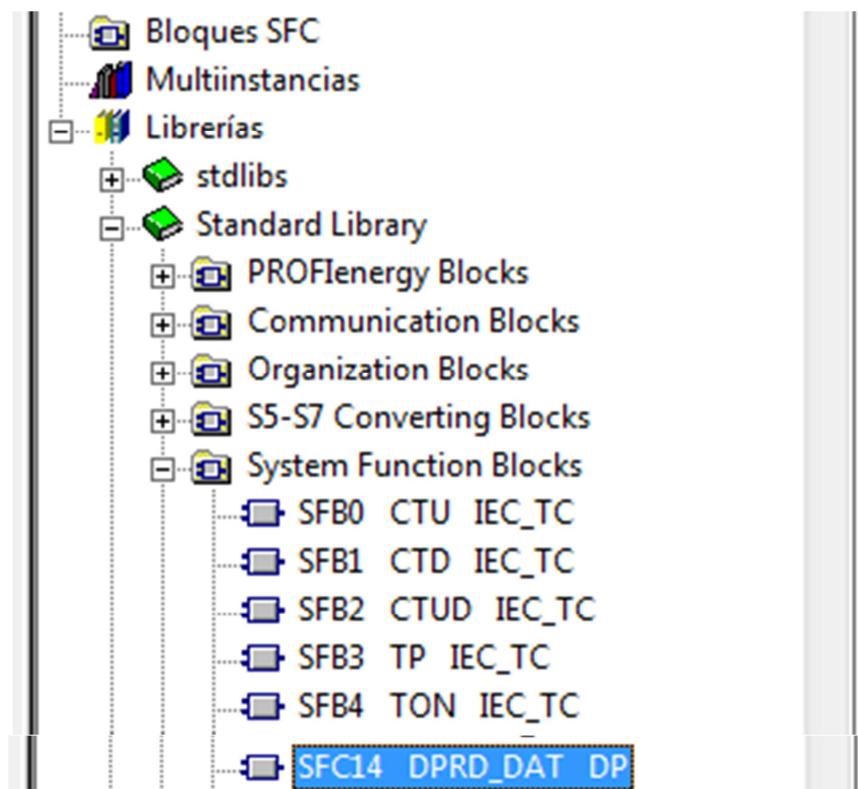


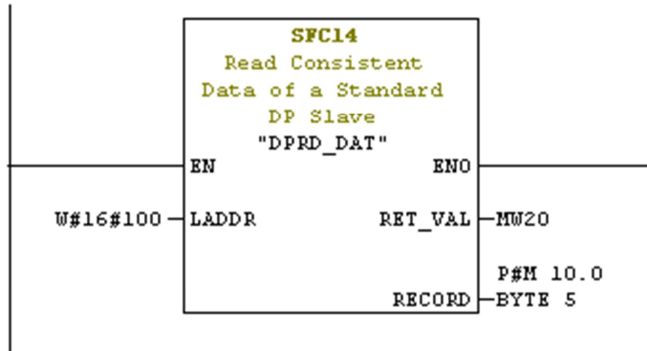
Figura 26: Detalle SFC 14



En System Function Blocks buscaremos la función SFC 14, y pinchando sobre ella, la arrastraremos hasta la interfaz de programación. La configuraremos como se muestra en la *Figura 27*:

**Segm. 7**: Lectura del sensor Sitrans PROBE LU

Insertamos la función SFC 14.



**Segm. 8**: Lectura del sensor Sitrans PROBE LU

Insertamos el bloque "MOVE" indicando a su entrada y salida la Marca 10 (donde guarda la función SFC 14 la lectura del sensor).

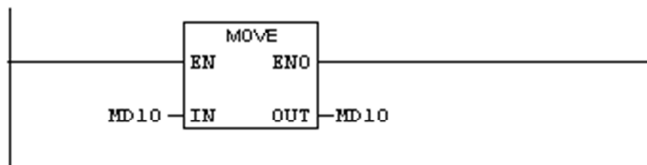


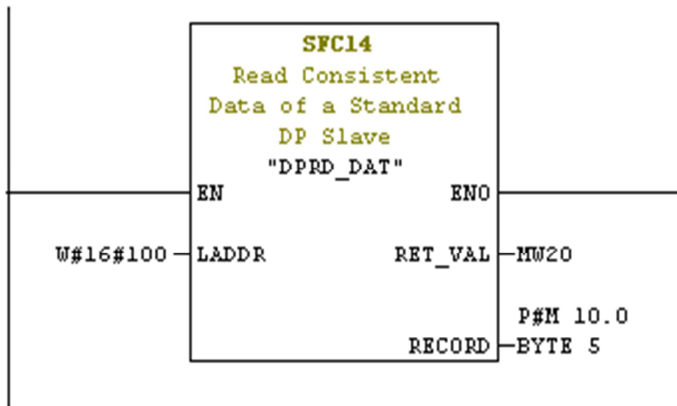
Figura 27: Función SFC 14 y bloque de transferencia MOVE

Si observamos la imagen anterior, podemos ver que la función SFC 14 guarda la lectura del sensor en el byte 5 de la marca 10 (salida **RECORD**). La entrada **LADDR** hace referencia a la dirección de salida del sensor y **RET\_VAL** nos indica si se ha cometido algún error y de qué tipo. Para poder visualizar los valores guardados en la marca 10 insertaremos un bloque MOVE como se ve en la imagen anterior.

El último paso será leer el valor del sensor. Pincharemos en **Observar sí/no**. Si nos fijamos en la función veremos lo siguiente:

**Segm. 7 :** Lectura del sensor Sitrans PROBE LU

Insertamos la función SFC 14.



**Segm. 8 :** Lectura del sensor Sitrans PROBE LU

Insertamos el bloque "MOVE" indicando a su entrada y salida la Marca 10 (donde guarda la función SFC 14 la lectura del sensor).

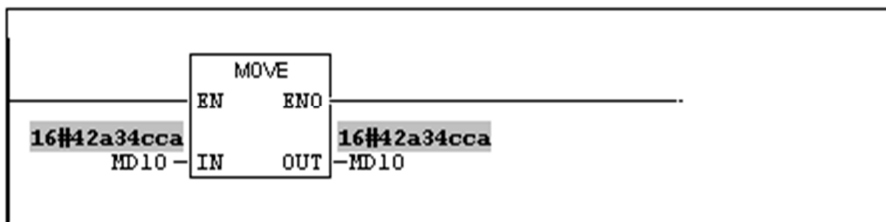
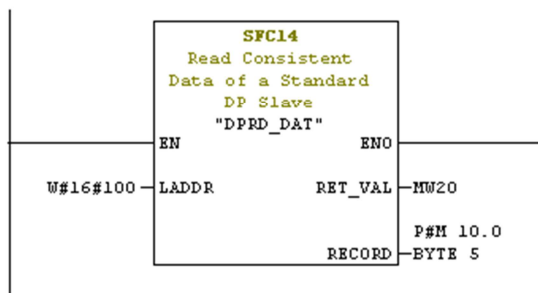


Figura 28: Lectura del sensor usando la función SFC 14

Como podemos ver, el valor leído es ininteligible. Para mostrar el valor leído de forma entendible, pincharemos sobre el con el segundo botón del ratón y en el menú desplegado, haremos como sigue: *Notación* → *Coma flotante*. De esta forma obtendremos el valor leído en notación decima con dos decimales como se ve en la imagen siguiente (Figura 29):

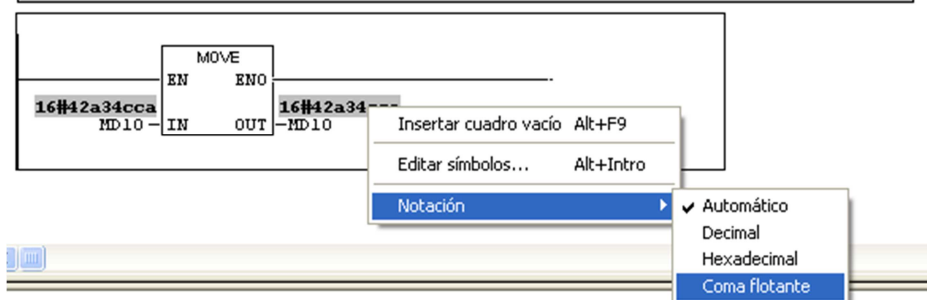
**Segm. 7:** Lectura del sensor Sitrans PROBE LU

Insertamos la función SFC 14.



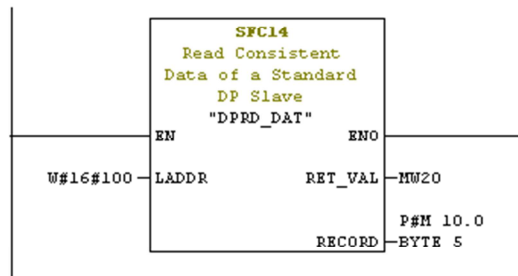
**Segm. 8:** Lectura del sensor Sitrans PROBE LU

Insertamos el bloque "MOVE" indicando a su entrada y salida la Marca 10 (donde guarda la función SFC 14 la lectura del sensor).



**Segm. 7:** Lectura del sensor Sitrans PROBE LU

Insertamos la función SFC 14.



**Segm. 8:** Lectura del sensor Sitrans PROBE LU

Insertamos el bloque "MOVE" indicando a su entrada y salida la Marca 10 (donde guarda la función SFC 14 la lectura del sensor).

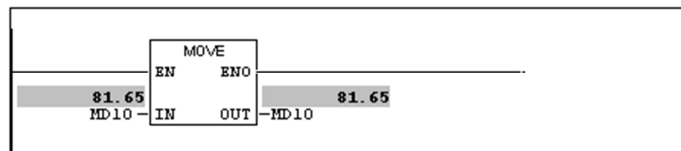


Figura 29: valor leído en notación decimal con dos decimales

## 6.6. Guardar y cargar el programa.

El último paso será guardar el programa y cargarlo en el autómata. Para ello primero desactivaremos **Observar sí/no** y pincharemos sobre el icono de **Guardar** confirmando la orden haciendo clic en **Sí**.

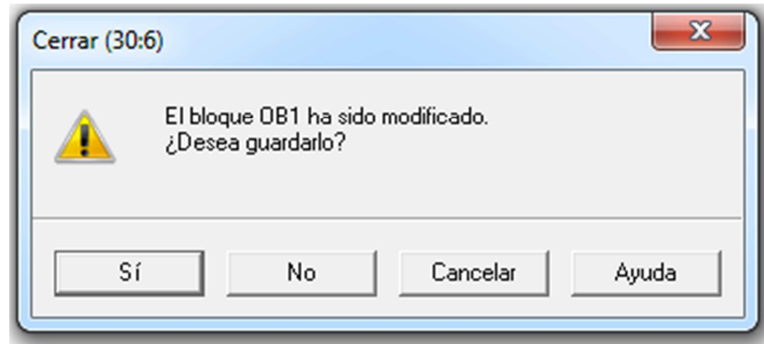


Figura 30: Guardar el programa

A continuación pincharemos sobre **Cargar** y en el mensaje que nos aparezca haremos clic en **Sí** para comenzar con la carga del programa.

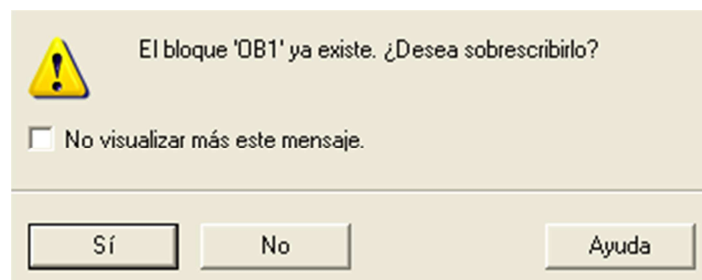


Figura 31: Cargar el programa

Hecho esto, cerraremos la interfaz de programación ya que en un principio no vamos a programar más y con el código que hemos cargado en el autómata será suficiente para la práctica.

Con estos pasos sencillos ya habremos guardado y cargado el programa. Lo siguiente que haremos será volver a la interfaz de trabajo del proyecto. Si nos fijamos detenidamente podemos observar que se ha incorporado la función **SFC 14**. Como ya se comentó con anterioridad, esta función es específica para leer los dispositivos pasivos en un sistema maestro-esclavo y además es una función del sistema, con lo cual, no podremos acceder a ella para modificarla.

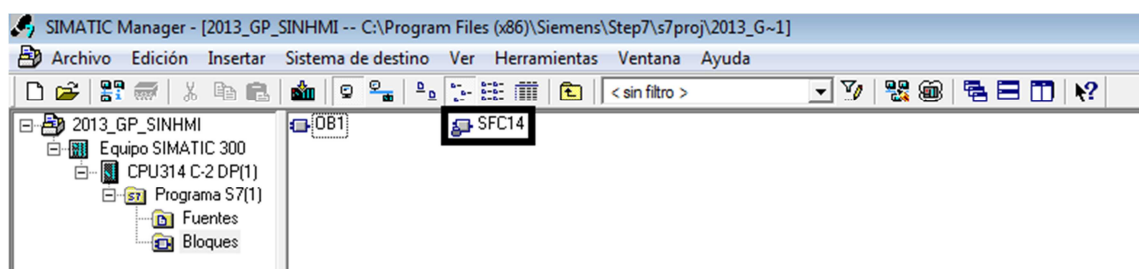


Figura 32: SFC 14 en la interfaz de trabajo

En la interfaz de trabajo, pincharemos sobre **Cargar**, y de nuevo, cargaremos todo el proyecto en el autómata.

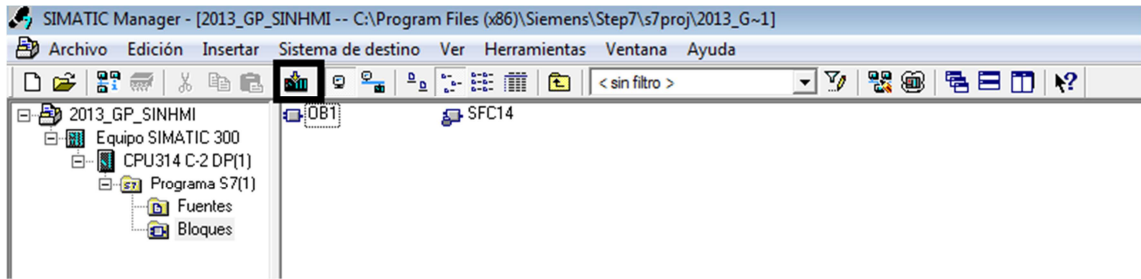


Figura 33 Cargar la configuración en el autómata

Seguiremos los pasos indicados para completar la carga de nuestro trabajo en el autómata como muestra la figura siguiente:

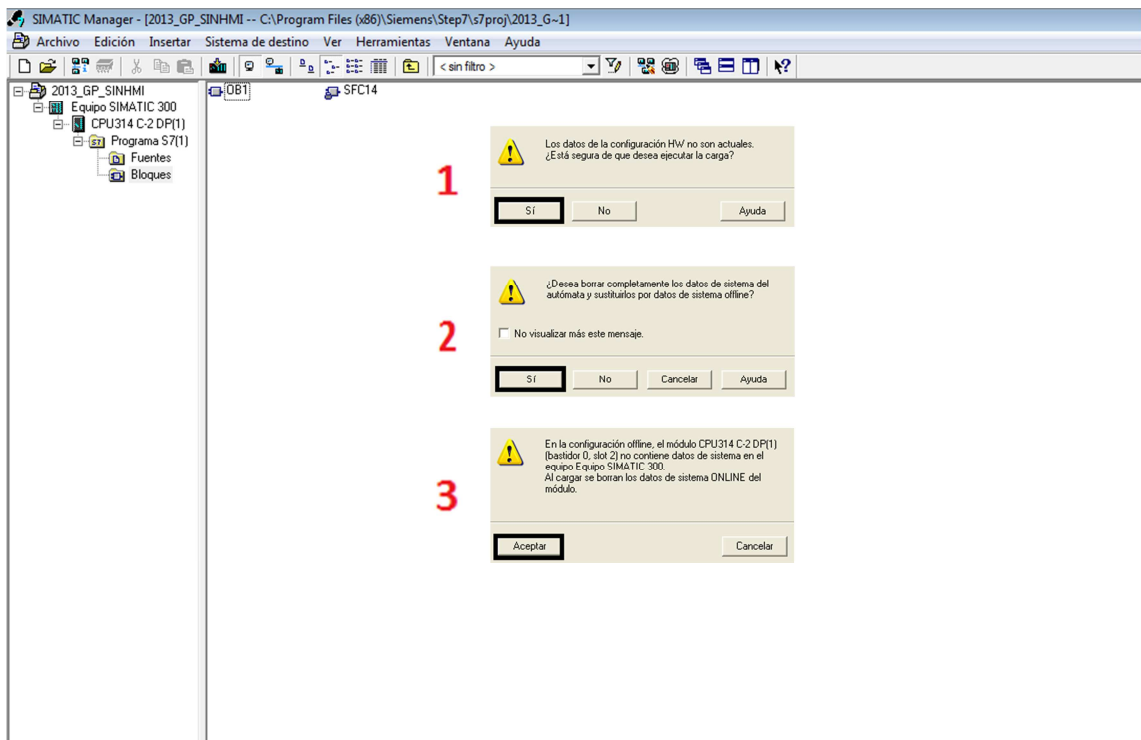


Figura 34: Confirmar la carga de datos

Después tendremos que volver a parar el autómata:

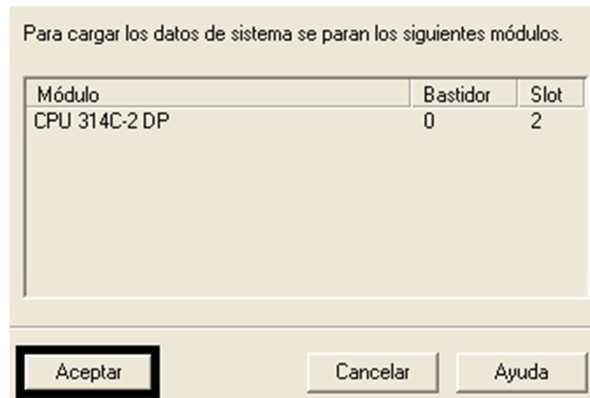


Figura 35: Parar el autómata

A continuación, sobrescribir el bloque de programación **OB1**:

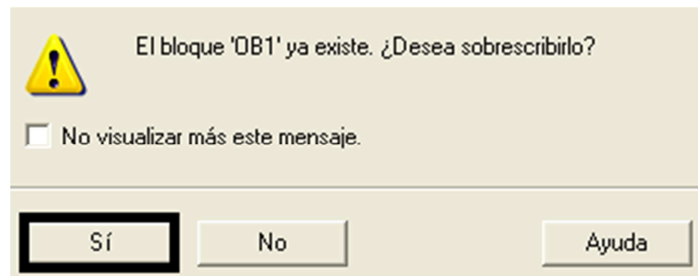


Figura 36: Sobrescribir el bloque de programación

Y por último, arrancar de nuevo la cpu del autómata, para finalizar así la carga de todo el trabajo realizado:

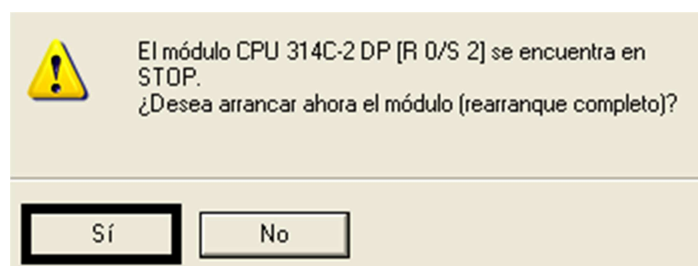
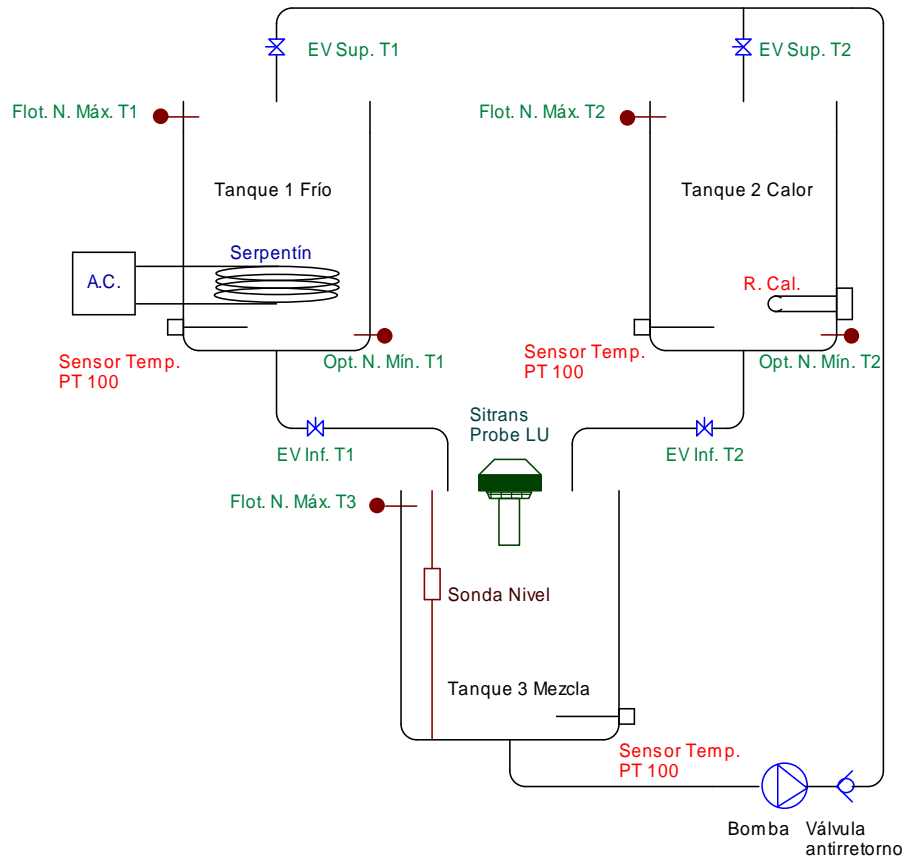


Figura 37: Arrancar el autómata

## 6.7. Programa todo/nada con E/S digitales.

Una vez que ya hemos comprobado las *E/S* digitales, tomado la lectura del sensor y guardado y cargado los programas, es el momento de configurar una aplicación sencilla todo/nada con las entradas y salidas digitales. El programa se encargará de controlar la maqueta desde las *E/S* digitales. Más adelante veremos algunos ejemplos integrando entradas y salidas analógicas.

En primer lugar vamos a representar una vez más el sistema físico a controlar. De esta forma podremos observar las necesidades y el funcionamiento de esta aplicación (*Figura 38*).



*Figura 38: Sistema físico a controlar*

Al tratarse de una aplicación digital, el grafcet será bien sencillo. En la imagen siguiente (Figura 39) podemos observarlo.

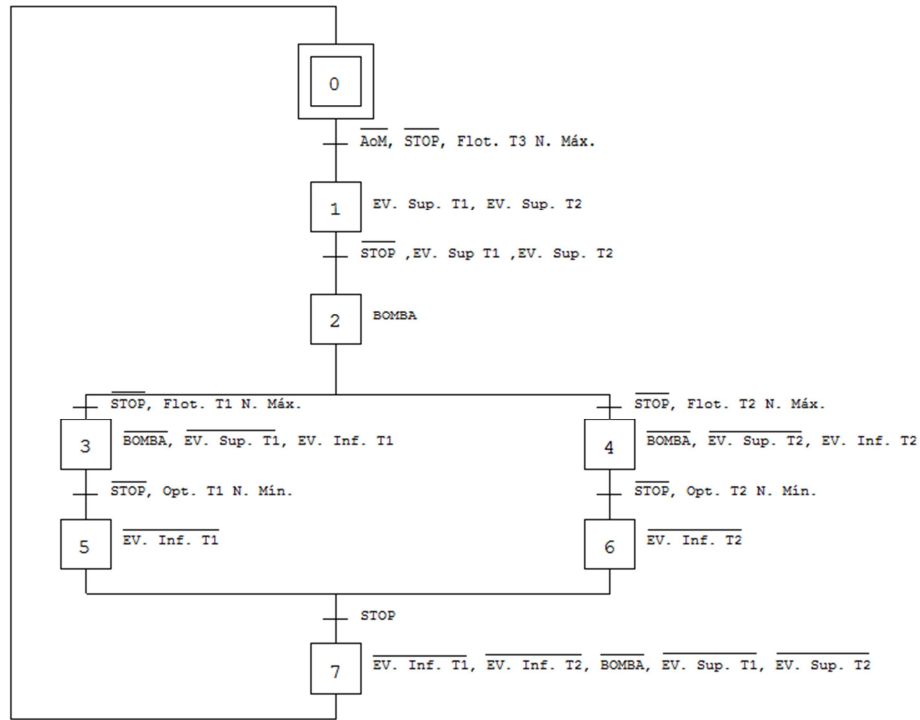


Figura 39: Grafcet

### Primera etapa.

El sistema comprobará que el modo Automático o Manual esté inactivo (en caso de estar activo, estaríamos en modo Manual y no arrancaría el programa). La parada de emergencia o STOP ha de estar inactiva también. En ese caso, si se activa el flotador-interruptor del tanque 3, se abrirán las electro-válvulas de los tanques superiores (EV. Sup. T1 y EV. Sup. T2). Esto se hace para cuando se arranque la bomba, el líquido tenga una vía de salida y no meta presión a las tuberías. La Figura 40 muestra lo descrito anteriormente:



Segm. 1 : Electroválvula superior del tanque 1.

Comprobamos que estamos en automático y la parada de emergencia está inactiva. Si se activa el flotador del tanque 3, abrimos las EV del tanque 1 y 2.

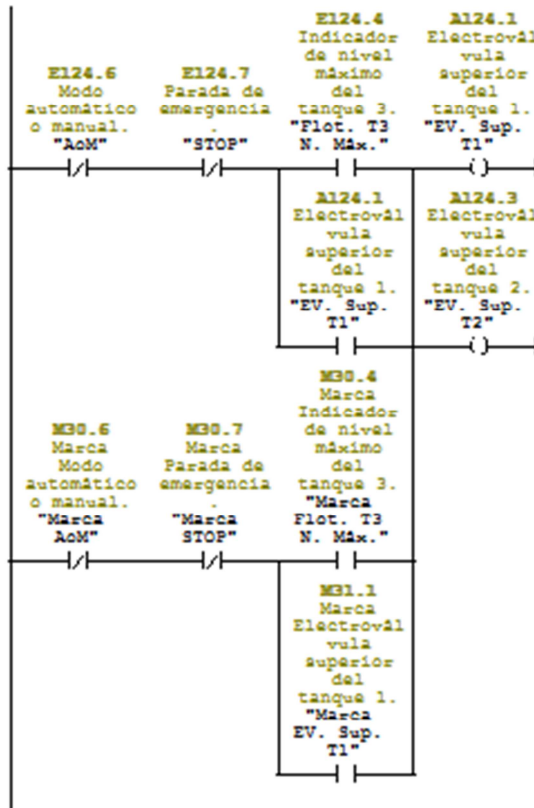


Figura 40: Primera etapa

## Segunda etapa.

A continuación, comprobamos que la parada de emergencia siga inactiva y las válvulas superiores de los tanques 1 y 2 estén abiertas. Entonces, arranca la bomba, tal y como muestra la siguiente imagen (Figura 41):

Segm. 2 : Bomba hidráulica del sistema.

Al abrir las EV's, arrancamos la bomba.

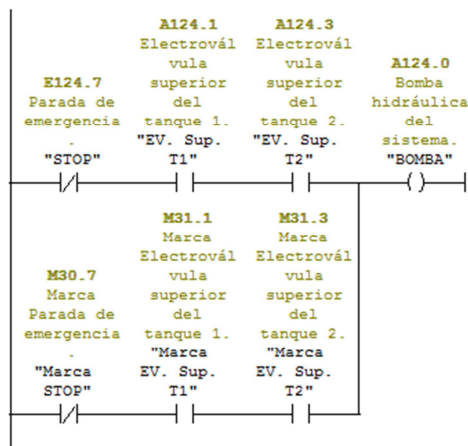
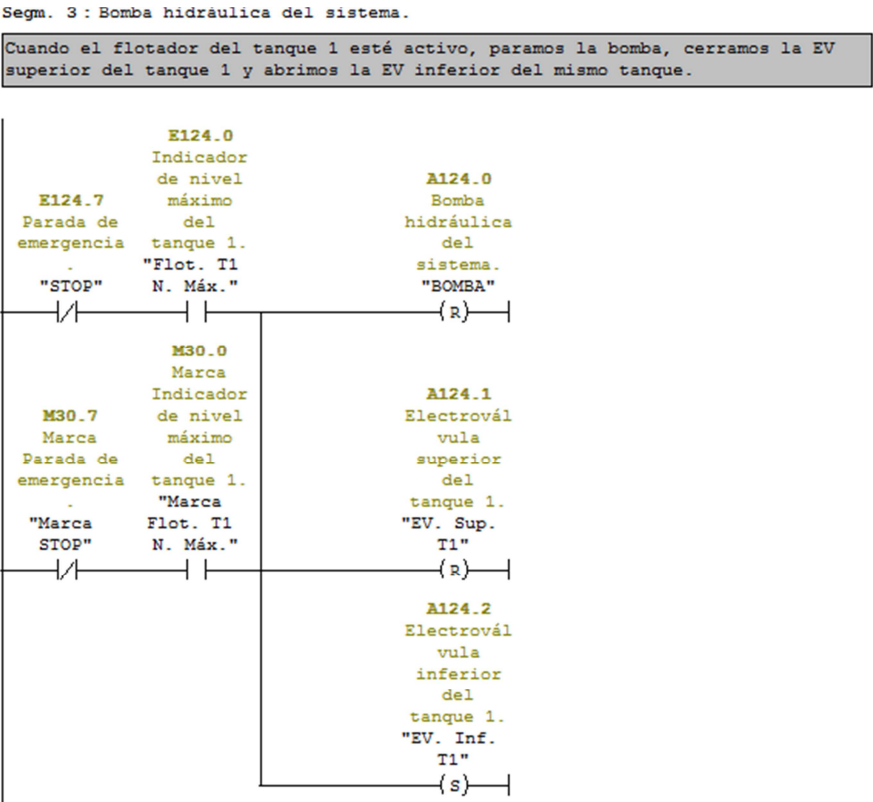


Figura 41: Segunda etapa

**Tercera etapa.**

Si la parada de emergencia sigue inactiva y se activa el flotador-interruptor (Flot. T1 N. Máx.) del tanque 1, esto significa que el depósito está lleno, por lo tanto paramos la bomba, cerramos la válvula superior y abrimos la válvula inferior (EV. Inf. T1) como se ve en la *Figura 42*:



*Figura 42: Tercera etapa*

**Cuarta etapa.**

Ídem a la tercera, aplicada al tanque 2. En la imagen a continuación (*Figura 43*) puede verse el código implementado.

Segm. 4 : Bomba hidráulica del sistema.

Cuando el flotador del tanque 2 esté activo, paramos la bomba, cerramos la EV superior del tanque 2 y abrimos la EV inferior del mismo tanque.

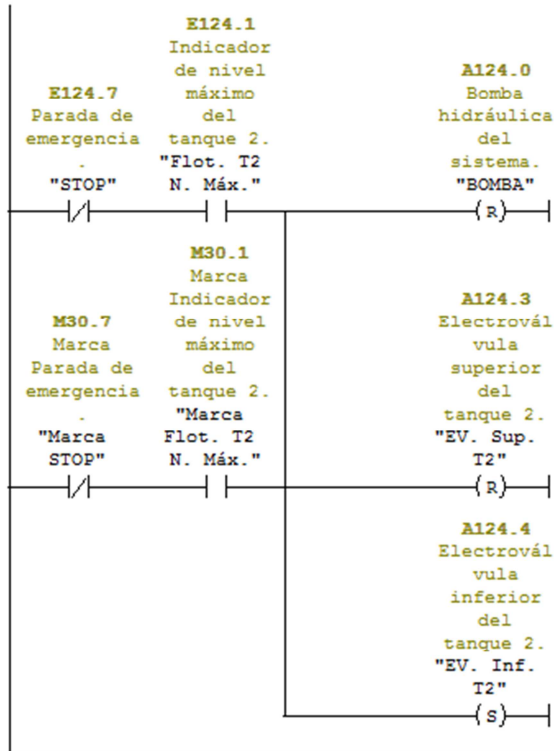


Figura 43: Cuarta etapa

### Quinta etapa.

Si la parada de emergencia, STOP, no está activa y se activa el sensor óptico del tanque 1 (Opt. T1 N. Mín.), esto significa que el depósito está casi vacío, por lo tanto, se cerrará el válvula inferior, EV. Inf. T1 como vemos en la *Figura 44*:

Segm. 5 : Electroválvula inferior del tanque 1.

Al activarse el sensor óptico del tanque 1, cerramos la válvula inferior del mismo tanque.

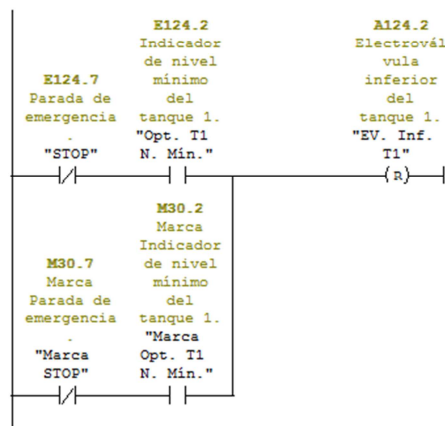
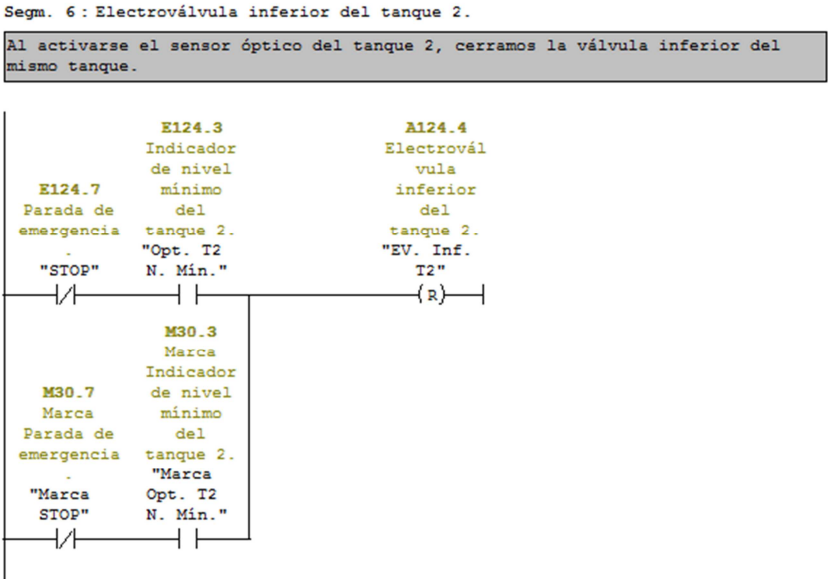


Figura 44: Quinta etapa

**Sexta etapa.**

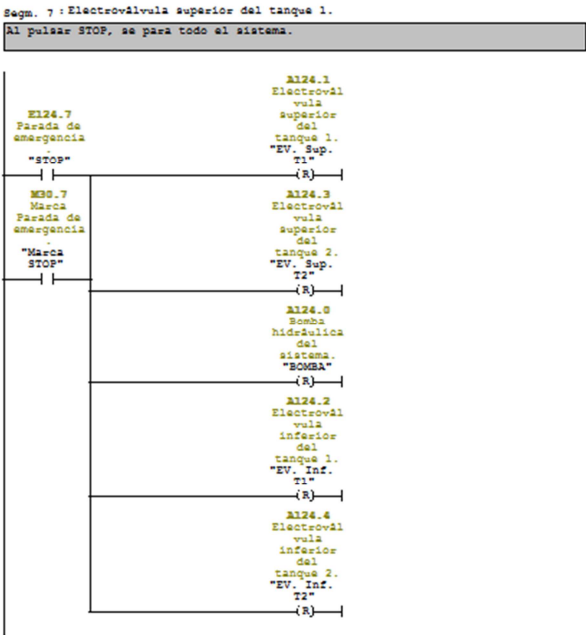
Igual que la quinta, aplicada al tanque 2. En la siguiente *Figura 45* vemos el código implementado.



*Figura 45: Sexta etapa*

**Última etapa.**

Para finalizar y como medida de seguridad, al activarse la parada de emergencia STOP, se desactivarán todas las electro-válvulas y la bomba tal y como vemos en la *Figura 46*:



*Figura 46: Última etapa*

Hay que mencionar la inclusión de marcas duplicando todas las entradas en nuestro código. Esto es necesario si queremos controlar el sistema desde la interfaz SCADA, ya que WinCC no puede activar o desactivar las E/S del autómeta.

## 6.8. Lectura y escritura de entradas y salidas analógicas.

Para leer o escribir una E/S analógica, primero necesitamos configurarla. Para ello, debemos volver a la configuración hardware y abrir las propiedades de las E/S analógicas. Pincharemos con el segundo botón sobre las entradas y salidas y seleccionaremos *Propiedades del objeto*, tal y como se ve en la siguiente imagen (Figura 47).

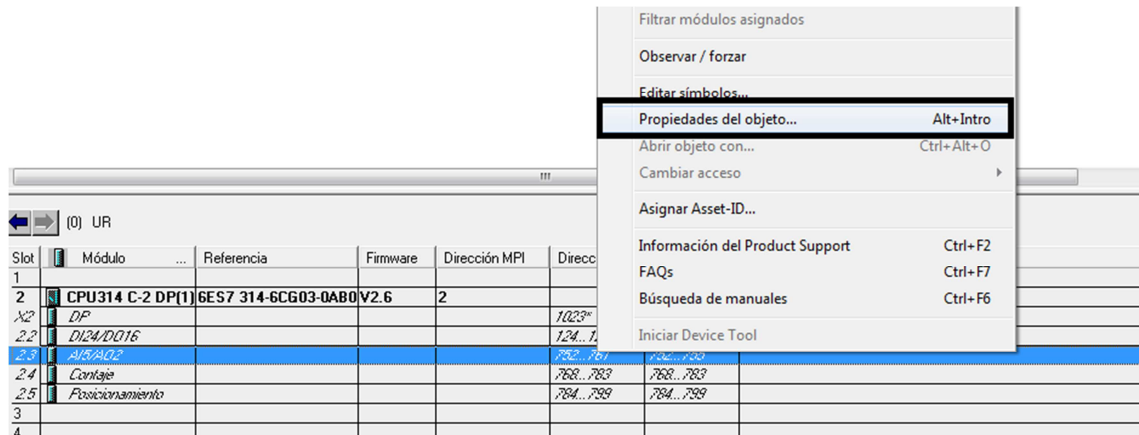


Figura 47: Acceder a las propiedades de las E/S analógicas

Al hacer esto, aparecerá la ventana de propiedades (Figura 48). En dicha ventana seleccionaremos la pestaña *Entradas* y configuraremos las entradas como muestra la imagen.

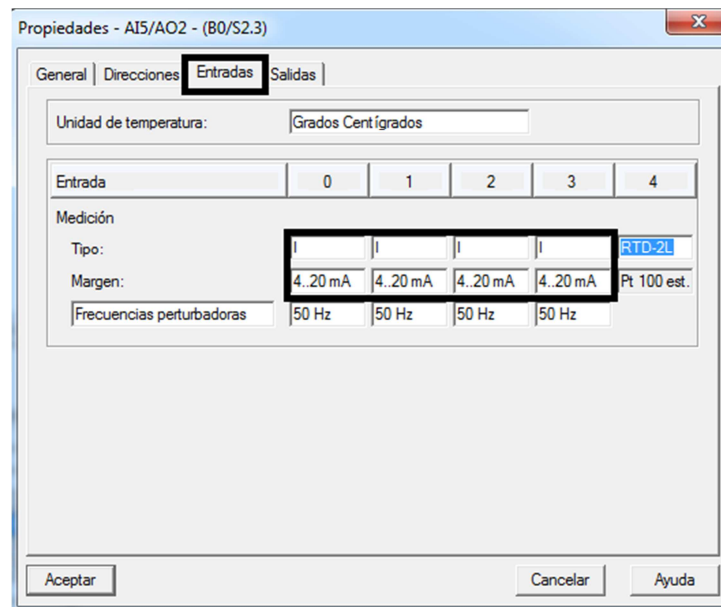
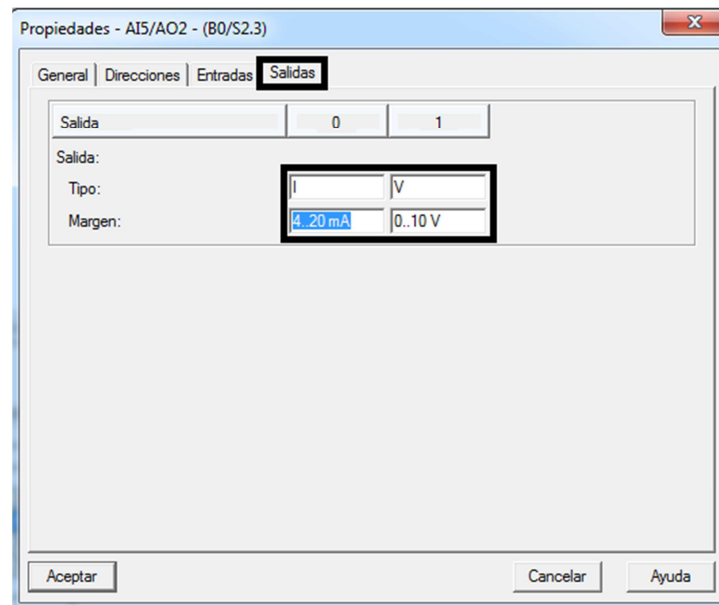


Figura 48: Propiedades de las entradas analógicas

Como ya sabemos, las entradas analógicas pueden conectarse tanto por tensión como por corriente. En nuestro caso, están conectadas por corriente, siendo transductores a dos hilos de 4 – 20 mA. La última entrada hace referencia a la termorresistencia que trae por defecto el módulo de *E/S* analógicas. Una vez estén configuradas, aceptaremos y procederemos de la misma forma con las salidas analógicas como vemos en la *Figura 49*:



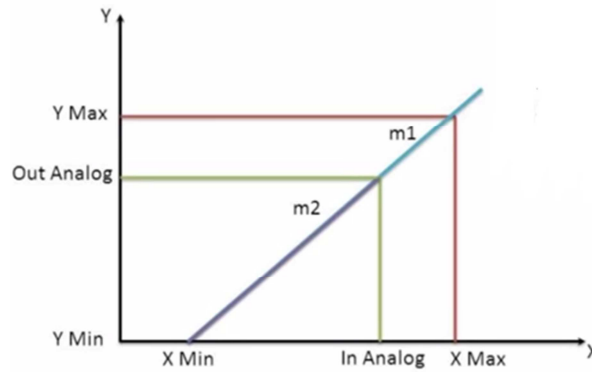
*Figura 49: Propiedades de las salidas analógicas*

Seleccionamos la pestaña *Salidas* y las configuramos como muestra la imagen anterior. El módulo de *E/S* analógicas del autómatas consta de dos salidas, una por corriente y otra por tensión. De este modo, hemos configurado la primera por corriente (4 – 20 mA) y la segunda por tensión (0 – 10 V).

El siguiente paso será programar una lectura y una escritura para una entrada y salida analógicas.

### 6.8.1. Lectura de una entrada analógica.

Por lo general, las entradas analógicas están conectadas a transductores de dos hilos que ofrecen una corriente de 4 a 20 mA. Para poder obtener el equivalente numérico a un valor de intensidad concreto entre ese rango, debemos escalar la señal de entrada. Para ello debemos obtener la pendiente de la recta que une dos puntos, tal y como muestra la siguiente *Figura 50*:



$$m1 = m2$$

$$\frac{Y \text{ Max} - Y \text{ Min}}{X \text{ Max} - X \text{ Min}} = \frac{Out \text{ Analog} - Y \text{ Min}}{In \text{ Analog} - X \text{ Min}}$$

$$Out \text{ Analog} = \frac{Y \text{ Max} - Y \text{ Min}}{X \text{ Max} - X \text{ Min}} * (In \text{ Analog} - X \text{ Min}) + Y \text{ Min}$$

Figura 50: Escalado de una entrada analógica

Normalmente, las señales analógicas utilizan el rango de 4 – 20 mA; por lo que ya conoceríamos el valor de X Min y X Máx. Por otro lado, si sabemos el rango de funcionamiento del transductor, ya tendríamos los valores máximos y mínimos de Y. Supongamos un sensor de temperatura que opera desde -50 a 100 grados. El valor mínimo de Y sería -50 y el valor máximo, 100. Si además conocemos la resolución del aparato, es decir, cuanto varía su salida conforme aumenta la temperatura 1°C, es fácil obtener la salida escalada.

Por otra parte, *Step 7* nos permite hacer esto de forma automática sin necesidad de realizar todas esas operaciones si utilizamos la función *FC 105*.

A la hora de implementar la función, deberemos buscarla en las librerías de nuestra interfaz de programación. El camino a seguir es el siguiente: *Librerías – Standard Library – TI-S7 Converting Blocks – FC 105 SCALE CONVERT* como vemos en la imagen siguiente (Figura 51):

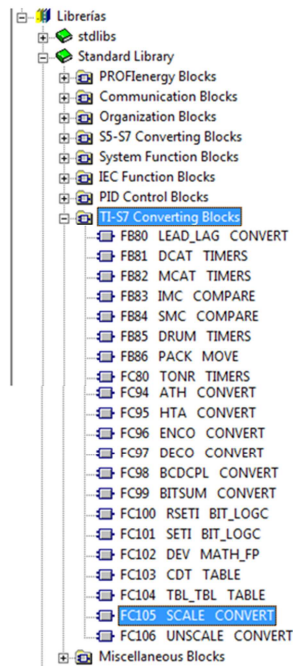


Figura 51: Acceso a la función FC 105

El siguiente paso será configurar la función adecuadamente. En la *Figura 52* podemos ver la función ya configurada.

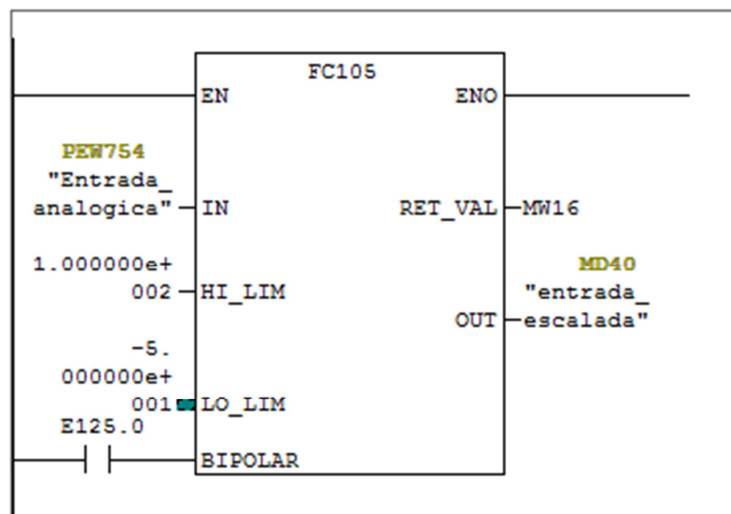


Figura 52: Función de escalado

Debemos tener en cuenta que en la entrada de la función estamos metiendo un dato de tipo entero, mientras que a la salida, tendremos un dato de tipo real. Por otro lado, ponemos a uno el parámetro *BIPOLAR* porque nuestro rango tomaría valores tanto negativos como positivos. Si nos moviéramos desde cero a valores positivos, la entrada *BIPOLAR* debería estar negada.



### 6.8.2. Escritura de una salida analógica.

Para la escritura analógica, utilizamos la función *FC 106 Unscale*. A la función se accede de la misma forma que a la función *FC 105*. La función *Unscale* realiza la operación inversa, es decir, toma un valor real y devolverá un entero. Por ello a la hora de utilizar esta función, primero deberemos transformar el valor entero que queramos meter en su entrada, a real. Se hace por tanto necesario transformar un entero en doble entero y posteriormente, ese doble entero, a real. Esto se hace como muestra la *Figura 53*:

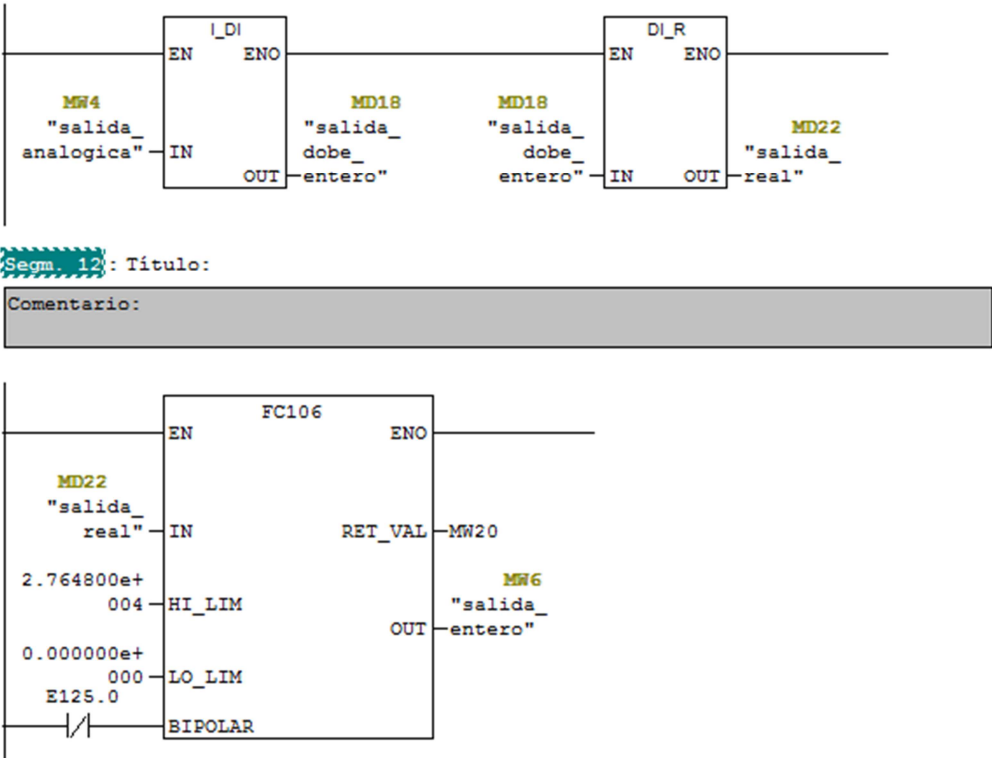


Figura 53: Escritura salida analógica

# Capítulo 7: Programación SCADA.

Este capítulo tratará sobre la elaboración de la interfaz hombre-máquina para el control del sistema físico, así como la realización de una serie de prácticas para el alumnado. Estas prácticas consistirán en monitorizar, supervisar y controlar los ejemplos realizados en el capítulo anterior.

La herramienta utilizada para esta tarea es el software *WinCC flexible*, versión 2008 y service pack 3. Se ha elegido este programa al ser muy intuitivo facilitando así su manejo.

## 7.1. El entorno de WinCC flexible.

Una vez más, conocer el entorno en el que vamos a trabajar, se convierte en algo sumamente importante. Por lo tanto, antes de empezar con la programación, vamos a conocer la interfaz de programación con un par de pinceladas, que no obstante, resultarán muy útiles a la hora de trabajar. Como ocurriera en el capítulo anterior, para saber más sobre la interfaz se pueden consultar los anexos de prácticas y manuales.

Cuando abrimos el programa, aparecerá la siguiente interfaz, mostrada a continuación en la *Figura 1*:

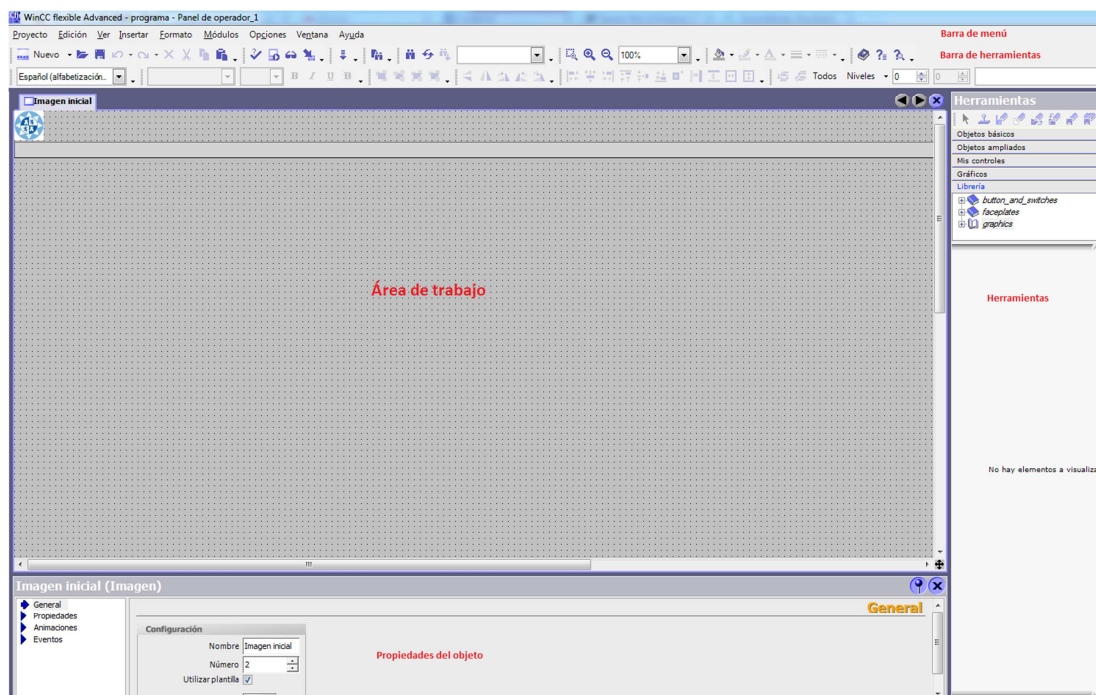


Figura 1: Interfaz de inicio de WinCC

La interfaz de *WinCC flexible* difiere de la interfaz de *Step 7*. Aún así, podemos definir varias zonas que explicamos a continuación:

- Barra de menú: muestra las acciones que podemos realizar en el proyecto y con los objetos del mismo.
- Barra de herramientas: proporciona un acceso directo a las opciones destacadas de la barra de menú.
- Área de trabajo: donde configuramos cada interfaz del proyecto.
- Herramientas: contiene todos los objetos y elementos para diseñar las interfaces.
- Propiedades del objeto: aquí editamos las propiedades de cada objeto o elemento que componen las interfaces.

El siguiente paso será adecuar la interfaz para poder trabajar de la mejor manera posible. Aquí radica la mayor diferencia con la interfaz de *Step 7*.

### 7.1.1. Adecuar la interfaz.

Para poder trabajar con garantías en WinCC flexible, se hace necesario desplegar una serie de elementos en la interfaz que nos proporcionarán ayuda e información sobre el trabajo que se está realizando. El primer paso será desplegar la *Vista del proyecto*, que se encuentra en el margen izquierdo, tal y como vemos en la *Figura 2*.

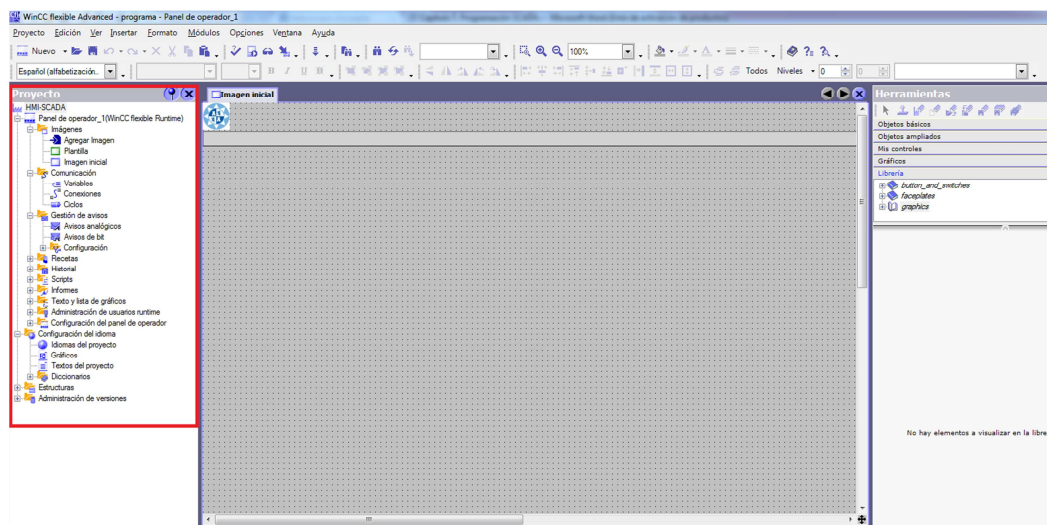


Figura 2: Desplegar la Vista del proyecto

Esta vista nos proporcionara las herramientas necesarias para configurar alarmas, avisos, crear nuevas imágenes, etc.

A continuación, abriremos la *Vista del objeto*. Para ello, iremos a la *Barra de menú*, pincharemos en *Ver* y seleccionaremos la vista necesaria (*Figura 3*).

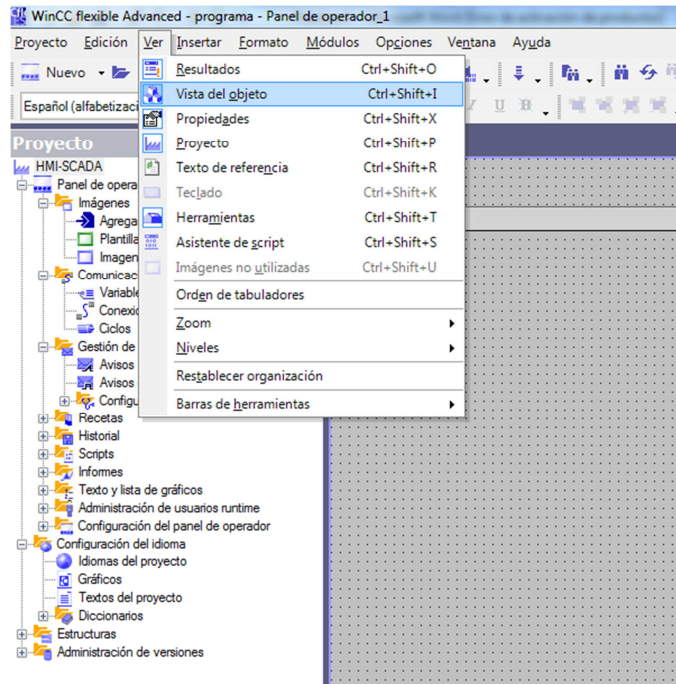


Figura 3: Abrir Vista del objeto

Por último, colocaremos la vista donde más fácil nos resulte acceder a ella en nuestra interfaz (Figura 4).

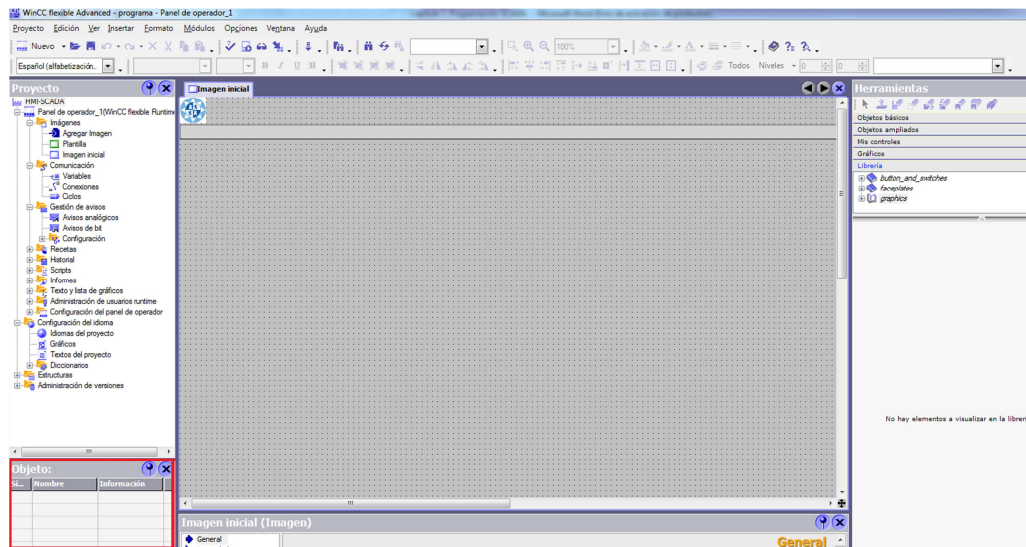


Figura 4: Vista del objeto

Con esta vista podremos acceder directamente a los objetos que configuremos en cada imagen de nuestro trabajo.

## 7.2. Configurar la conexión en WinCC flexible.

Ahora que ya conocemos la interfaz en la que vamos a trabajar y los elementos que la componen, es el momento de aprender a configurar una conexión en *WinCC flexible*. La comunicación se hará entre el autómatas y *WinCC flexible RunTime*, que nos servirá para visualizar y controlar nuestro sistema. La comunicación es un aspecto muy importante en cualquier tipo de sistema de control, por lo que pondremos mucho cuidado y atención a la hora de configurar para que todos los parámetros sean coherentes. Pero antes, debemos realizar la configuración hardware, como ya sucediera con *Step 7*.

### 7.2.1. Configuración Hardware.

El siguiente paso será realizar la configuración hardware de la estación HMI, de la misma forma que sucediera en la Práctica 1 con el resto de elementos que componen el sistema de control.

Para acceder a la configuración hardware, vamos a la interfaz de trabajo y pinchamos sobre **SIMATIC HMI-Station** en el menú jerárquico de la izquierda. Seguidamente haremos doble clic sobre **Configuración** como se ve en la *Figura 7*:

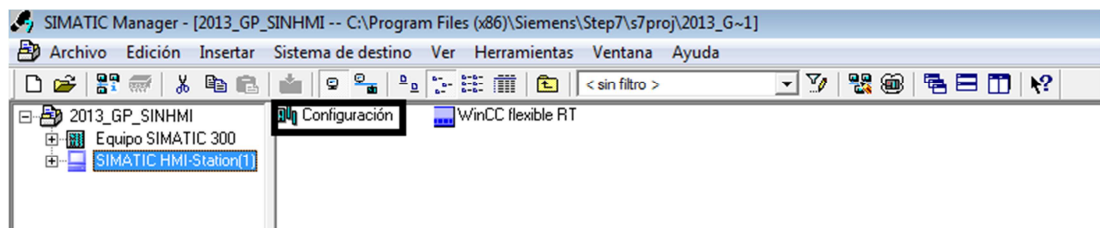


Figura 7: Abrir Configuración Hardware

Al hacer esto, abriremos la configuración de la estación hombre-máquina apareciendo la siguiente ventana (*Figura 8*).

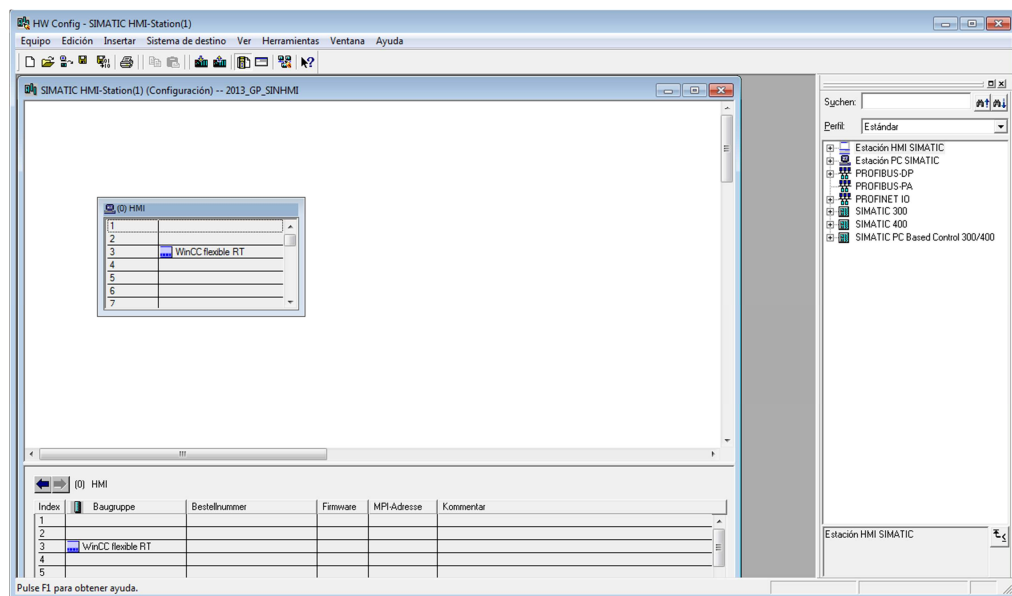


Figura 8: Configuración Hardware

El siguiente paso será agregar la tarjeta de comunicación a nuestra configuración. Para ello vamos a la parte derecha de la ventana, y en los elementos disponibles, seleccionamos **Estación HMI SIMATIC**. Desplegamos el menú y abriremos la carpeta **CP-PROFIBUS**. A continuación abrimos la carpeta **CP-5611** y pinchamos sobre la versión **SW V6.0 SP4** como muestra la *Figura 9*:

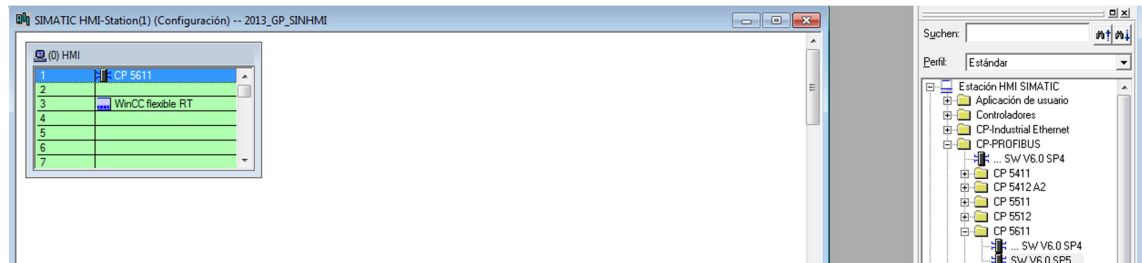


Figura 9: Tarjeta CP-5611

Como vemos, al pinchar sobre la tarjeta, se colorea en verde los lugares del bastidor en los que podemos colocarla. A continuación, pinchamos y arrastramos la tarjeta hacia el bastidor. Cuando lo hayamos hecho, deberemos conectarla a una red y ajustar los parámetros de ésta (*Figura 10*).

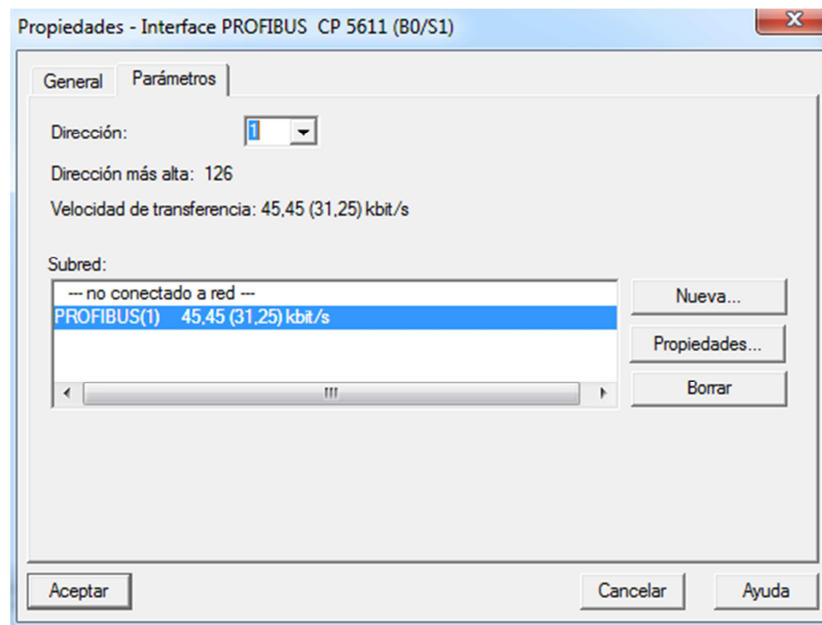


Figura 10: Ajustar los parámetros de la red

A la hora de configurar la red, ajustaremos la dirección a 1 y seleccionaremos la red **PROFIBUS**. Si observamos con detenimiento la imagen anterior (*Figura 10*), podemos ver que **no ha sido necesario crear ninguna red. Ésta ya fue creada con el asistente HW Config cuando ajustamos la configuración hardware del sistema en la Práctica 1**. A continuación pincharemos sobre **Aceptar**, insertando así nuestra tarjeta de comunicación, tal y como muestra la *Figura 11*:



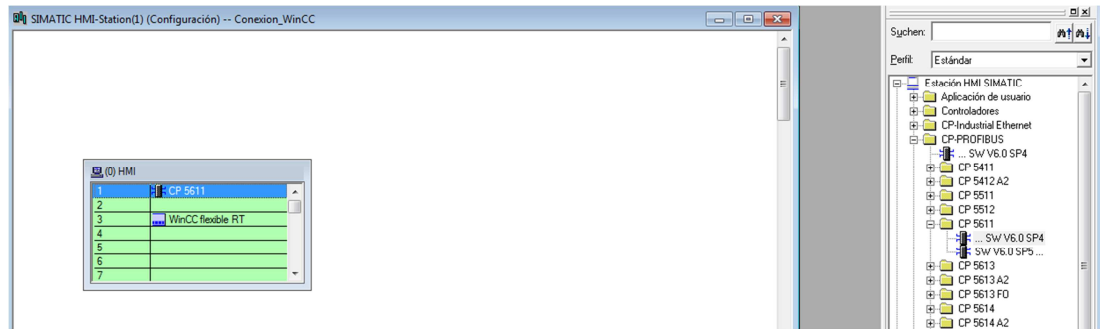


Figura 11: Tarjeta CP-5611 insertada en el bastidor

El siguiente paso será Guardar la configuración y cerrar el asistente. De esta forma habremos indicado qué interfaz de comunicación tenemos y a qué red se encuentra conectada. Esto es muy importante a la hora de configurar nuestras conexiones en *WinCC flexible*. Si observamos la interfaz de trabajo, veremos que aparece la tarjeta *CP-5611* (Figura 12).

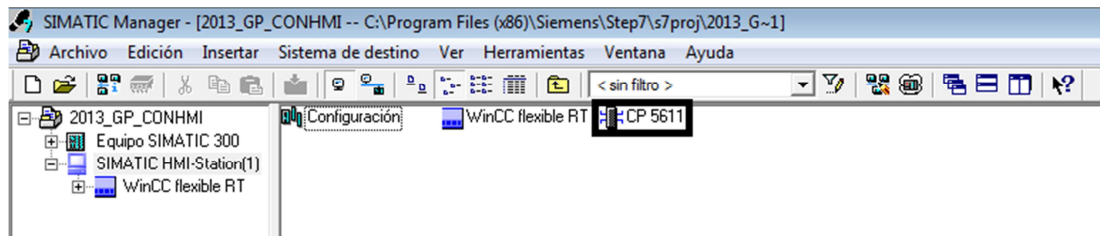


Figura 12: Tarjeta CP-5611 agregada a la interfaz de trabajo

Ahora sí, ya podemos configurar una conexión. Para abrir el configurador de conexiones de *WinCC flexible* iremos a la **vista del Proyecto** y haremos doble clic sobre **SIMATIC HMI Station**. En el menú desplegable que aparezca, repetiremos la misma operación sobre la carpeta **Comunicación**. Por último haremos doble clic nuevamente sobre **Conexiones** accediendo al configurador de conexiones, como se muestra en la Figura 13:

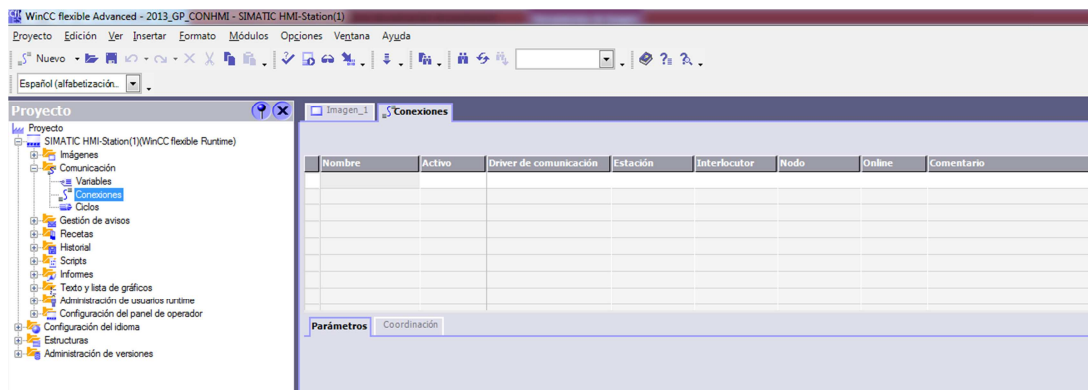


Figura 13: Abrir el configurador de conexiones

A continuación crearemos y configuraremos una conexión. Para ello haremos doble clic sobre cualquier lugar de la zona sombreada, creando así una conexión. Si quisiéramos crear más conexiones, repetiríamos el proceso en cualquier recuadro en blanco como muestra la *Figura 14*:

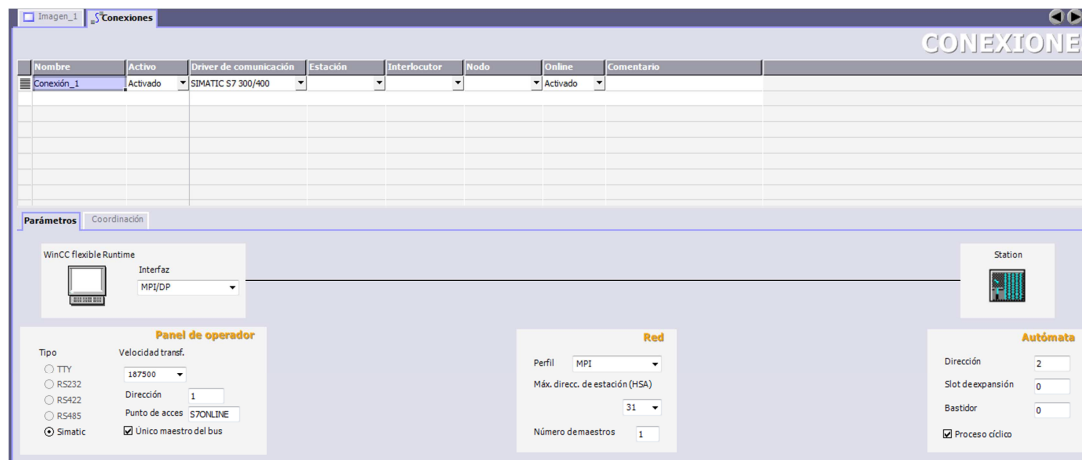


Figura 14: Crear una conexión

Por último, tendremos que configurar los parámetros de la conexión. Los parámetros debemos ajustarlos en consonancia con la red a la que hemos conectado nuestra interfaz de comunicación (tercer punto de este documento).

Como podemos ver, el configurador de conexiones de *WinCC* se divide en dos partes. En la parte superior, indicaremos qué tipo de CPU tenemos mientras que en la parte inferior, se indican los parámetros de conexión de la red, tales como la velocidad de transferencia, el perfil de red, etc.

En la parte superior, debemos indicar qué estación tenemos. Pincharemos en la pestaña de *Estación* y seleccionaremos la opción **PROFIBUS** → **\2013\_GP\_CONHMI\Equipo SIMATIC 300**. En este apartado indicamos desde dónde accedemos al autómata. En nuestro caso, accederemos desde el PC (*Figura 15*).

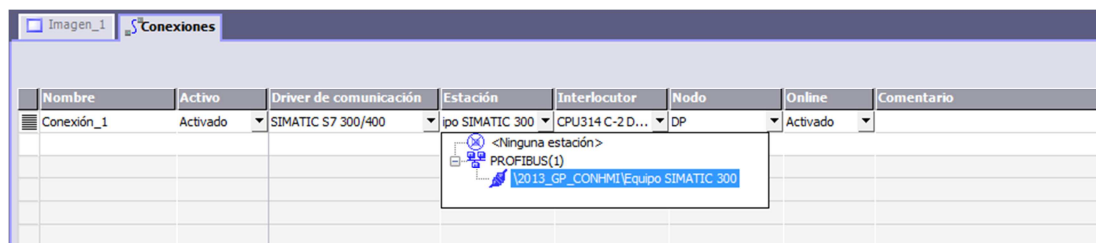


Figura 15: Seleccionar estación

Al hacer esto podremos observar (*Figura 16 a*)) como los demás campos se han rellenado de forma automática, quedando de esta manera:

- **Interlocutor:** CPU 314C-2DP. El autómata con el que vamos a trabajar.
- **Nodo:** DP. El perfil de la red.



Nombre	Activo	Driver de comunicación	Estación	Interlocutor	Nodo	Online	Comentario
Conexión_1	Activado	SIMATIC S7 300/400	\Conexion_Wi...	CPU 314C-2 DP	DP	Activado	

Figura 16 a): Configuración de la conexión

Por otro lado, la parte inferior se configurará de forma automática (Figura 16 b)), siendo sus parámetros, los mismos que los de la red con la que estamos trabajando (segundo punto de este documento), quedando de la forma:

- **Interfaz:** CP 5611
- **Panel de operador:**
  - **Velocidad de transferencia:** <sup>1</sup>45450 bits/s
  - **Dirección:** 1
  - **Punto de acceso:** S7ONLINE
  - **Tipo:** Simatic, marcando la casilla “Único maestro del bus”
- **Red:**
  - **Perfil:** DP
  - **Máxima dirección de estación (HSA):** <sup>2</sup>126
  - **Número de maestros:** 0
- **Autómata:**
  - **Dirección:** 2
  - **Slot de expansión:** <sup>3</sup>2
  - **Bastidor:** 0

Figura 16 b): Configuración de la conexión

<sup>1</sup> Velocidad limitada por PROFIBUS PA.

<sup>2</sup> Podemos ajustar la dirección más alta desde la interfaz PG/PC y al configurar una red nueva.

<sup>3</sup> No tiene por qué ser este valor. Puede ser 0 o cualquier otro.

### 7.3. Asociación de variables.

Una vez que ya hemos configurado nuestra conexión, el siguiente paso será asociar las variables, que configuramos en nuestra tabla de símbolos con *Step 7*, en *WinCC flexible*.

Vamos a explicar la asociación de variables de forma práctica a modo de ejemplo. Como ya se ha explicado en la introducción de esta práctica, vamos a realizar una interfaz sencilla. Esta interfaz constará de dos variables; una para la bomba del sistema y otra para el sensor de nivel del tanque 3 de la maqueta del laboratorio. Cabe añadir que el proceso de asociar variables es el mismo sin importar el número de éstas que utilicemos en nuestro proyecto.

Por tanto, el primer paso será crear la variable. Para ello abriremos el configurador haciendo doble clic sobre la carpeta **Comunicación**, y seguidamente haremos lo propio sobre **Variables** en el menú jerárquico que aparece en la **vista del Proyecto**, tal y como muestra la *Figura 17*.

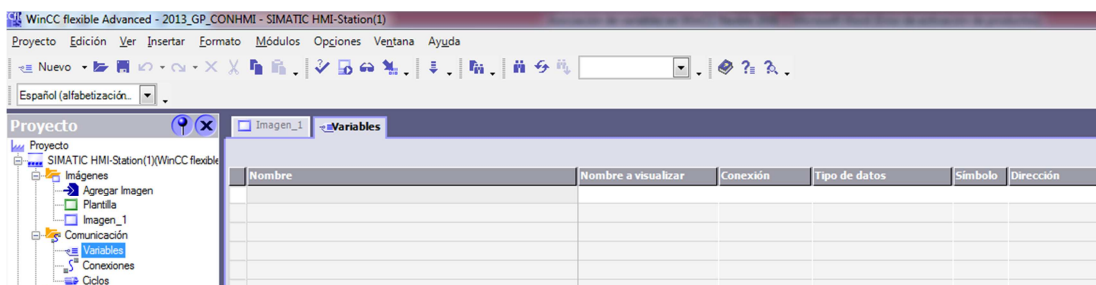


Figura 17: Configurador de variables

Y a continuación, simplemente tendremos que hacer clic sobre la zona sombreada, apareciendo así la variable. Obsérvese que el nombre de la misma es *Variable\_1* y que no está asociada a ninguna conexión (*Figura 18*).



Figura 18: Nueva variable

El siguiente paso será asociar esta variable de *WinCC* a otra perteneciente a nuestra tabla de símbolos de *Step 7*. Para ello, pinchamos en la pestaña del campo **Símbolo** y en la jerarquía desplegable haremos doble clic como sigue: **Proyecto** → **Nombre del proyecto** (2013\_GP\_CONHMI en nuestro caso) → **Equipo SIMATIC 300** → **CPU 314C-2 DP** → **Programas S7** → **Symbols**. Como muestra la *Figura 19*, observamos la tabla de símbolos creada con el programa *Step 7*. Elegiremos la variable **Bomba** y pincharemos sobre el icono de confirmar, remarcado en negro.

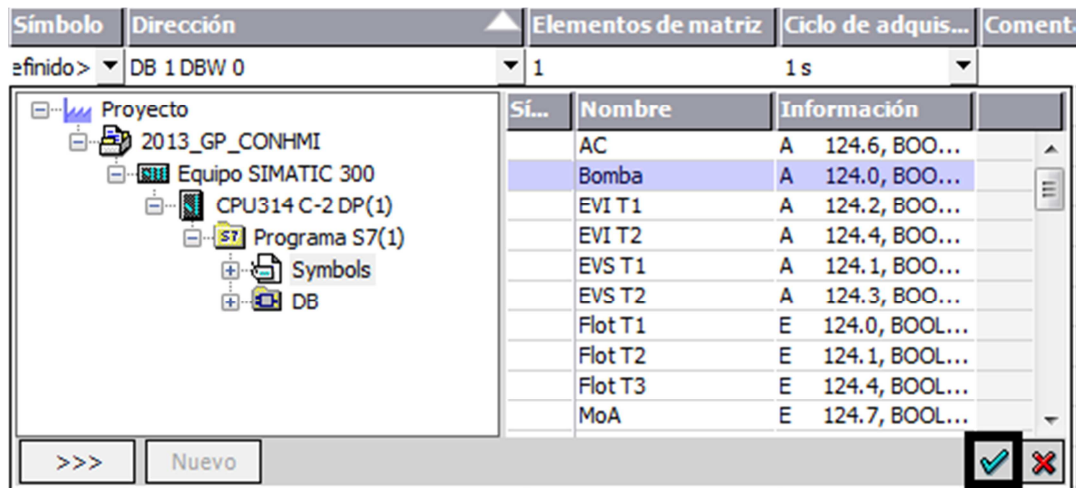


Figura 19: Lista de variables

De esta manera, ya habremos asociado las variables de *WinCC flexible* y *Step 7* en nuestro proyecto como muestra la *Figura 20*:

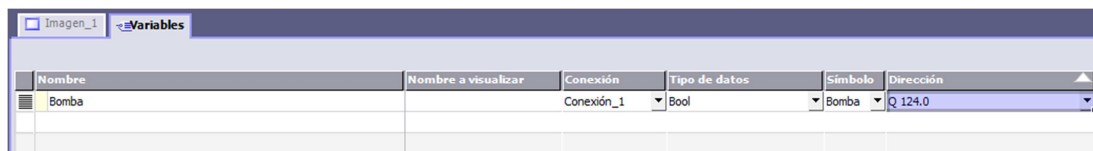


Figura 20: Variables asociadas

Si observamos la imagen anterior (*Figura 20*) con detenimiento, podremos observar que los parámetros de nuestra variable en *WinCC* han cambiado a los parámetros de la variable asociada de *Step 7*. Recordemos que cuando hemos creado la variable, ésta carecía de nombre (*Variable\_1*) y no estaba conectada a ninguna conexión. Ahora está asociada a la conexión que hemos configurado en el punto anterior que será con la que trabajemos.

Por último, repetiremos los mismos pasos para crear la variable del sensor de nivel del tanque 3. Cuando accedamos a la tabla de símbolos, elegiremos la variable **Flot T3**. De esta forma ya habremos creado las dos variables necesarias para nuestro ejercicio tal y como puede verse en la *Figura 21*:

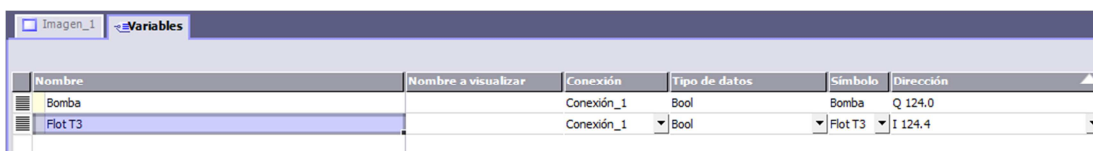


Figura 21: Variables creadas

Una vez hecho esto, no vamos a necesitar más el configurador de variables, por lo que podremos cerrarlo haciendo clic en el icono de **Cerrar** de la parte superior derecha.

**Nota 1:** no es necesario configurar ninguna conexión con anterioridad a la asociación de variables. Se puede crear una conexión posteriormente y asociarla a las variables deseadas.

**Nota 2:** se pueden crear variables internas en *WinCC*. Estas variables no ocuparán espacio en la memoria del autómatas, por lo tanto no aparecerán en la tabla de símbolos. Estas variables no están asignadas a ninguna conexión.

## 7.4. Crear las interfaces del SCADA.

Una vez que ya hemos configurado nuestra conexión y asociado las variables, es hora de crear las interfaces de nuestro sistema *SCADA*. El propio sistema ya ofrece algunas interfaces hechas por defecto, tales como las ventanas de avisos o la interfaz de administración de usuarios. No obstante, todas las interfaces por defecto, pueden ser creadas a partir de una imagen en blanco y añadiendo los elementos necesarios que aparecen en *Herramientas*.

Los objetos que aparecen en la zona de herramientas, son objetos sencillos de una apariencia muy básica. Si queremos tener acceso a elementos visualmente más complejos, deberemos cargar las librerías del sistema.

### 7.4.1. Cargar librerías.

Para cargar las librerías del sistema iremos a la zona de *Herramientas* y pincharemos sobre la pestaña *Librería*. En la barra de menú de la zona de Herramientas, pinchamos sobre el icono remarcado *Abrir una librería* como muestra la *Figura 22*:

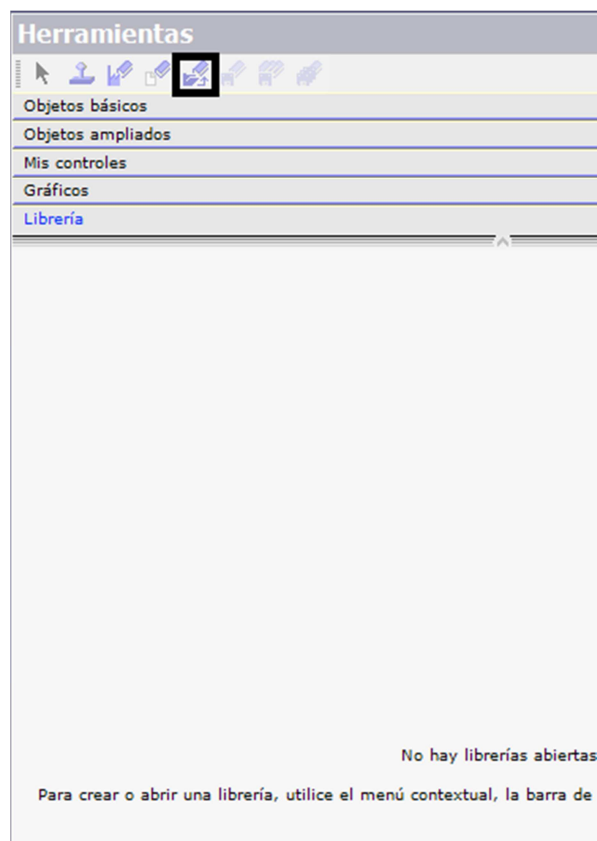


Figura 22: Abrir una librería

En la ventana que aparezca, seleccionaremos **Librerías del sistema** y a continuación, cargaremos las tres librerías que aparecen seleccionándolas y pinchando en **Abrir** (Figura 23). Este proceso hay que repetirlo tres veces, una por cada librería.

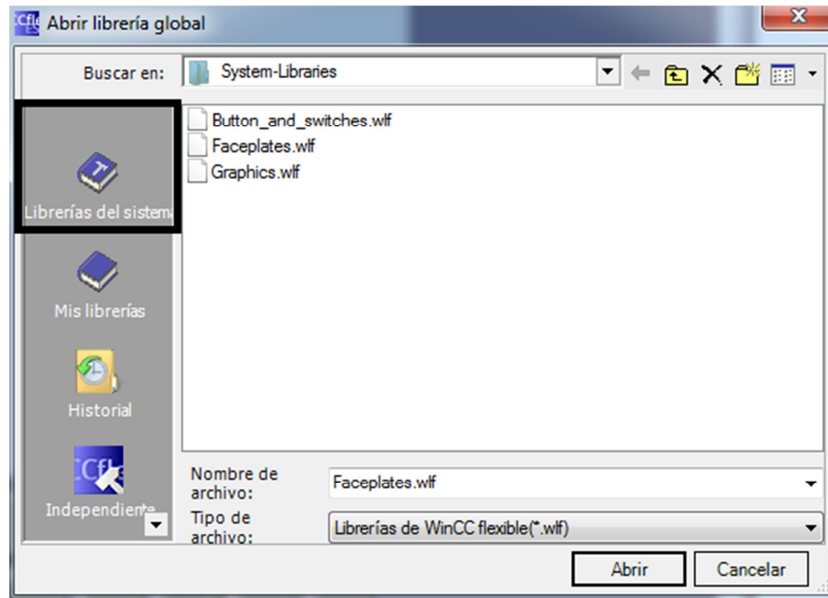


Figura 23: Cargar las librerías del sistema

De esta forma habremos cargado las librerías del sistema y las tendremos listas para ser utilizadas en la zona de Herramientas, tal y como muestra la Figura 24:

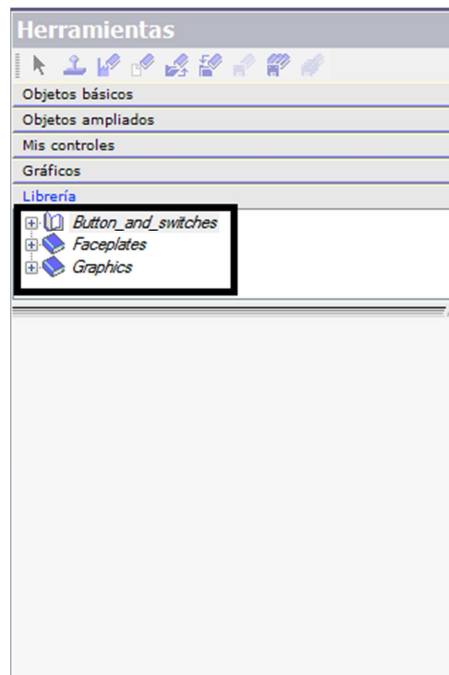


Figura 24: Librerías en la zona de Herramientas

Ahora tendremos un abanico más amplio de objetos para configurar las interfaces de nuestro sistema SCADA. El siguiente paso será, por tanto crearlas. Para ello, vamos a tomar como ejemplo la interfaz que muestra el proceso completo.

### 7.4.2. Interfaz central del proceso

Como ya hemos comentado, para crear cualquier interfaz, debemos recurrir a los objetos que figuren en las *Herramientas*. Una vez tenemos claro qué vamos a representar y cómo lo vamos a hacer, simplemente debemos seleccionar los objetos y arrastrarlos hasta la interfaz. En la imagen a continuación (*Figura 25*) se muestra la interfaz terminada:

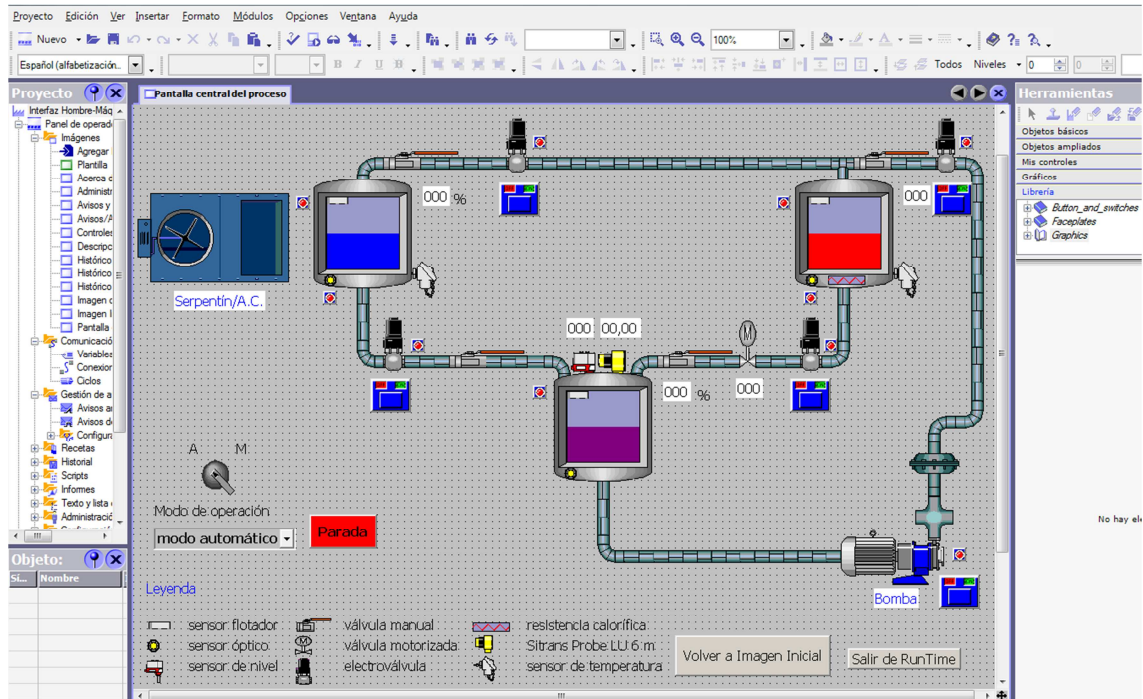


Figura 25x: Interfaz central del proceso

Como vemos, la interfaz intenta reproducir todos los elementos que actúan sobre el sistema físico. A continuación vamos a relatar qué elementos intervienen y cómo configurarlos.

#### Campos de entrada/salida.

Los campos de entrada/salida representarán el valor numérico de cualquier variable escalable, por ejemplo, el nivel. Para implementar un campo de entrada/salida iremos a las *Herramientas*, pincharemos en *Objetos básicos* y seleccionaremos *Campo E/S* como muestra la *Figura 26*:

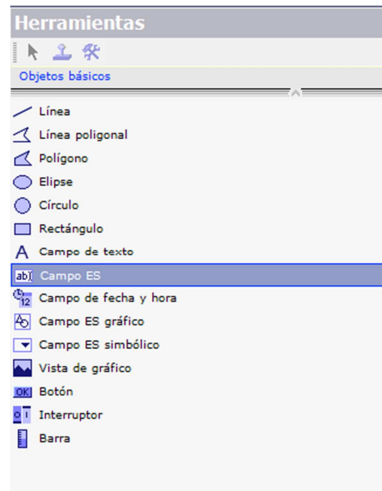


Figura 26: Implementar un campo de E/S

Si pinchamos sobre un campo de entrada/salida, podremos configurarlo según nuestras necesidades y asociarlo a una variable para que muestre su valor durante el proceso (Figura 27).

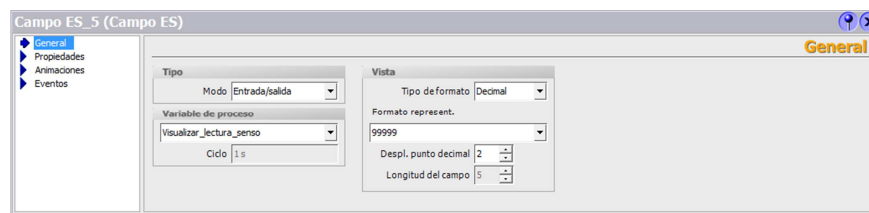


Figura 27: Propiedades del campo E/S

## Botones.

Realmente útiles a la hora de cambiar de interfaz, configurar paradas de emergencia o para salir del sistema de control y cambiar de una imagen a otra. Para configurar un botón deberemos ir a *Herramientas* y seleccionarlo en *Objetos básicos* (Figura 28).

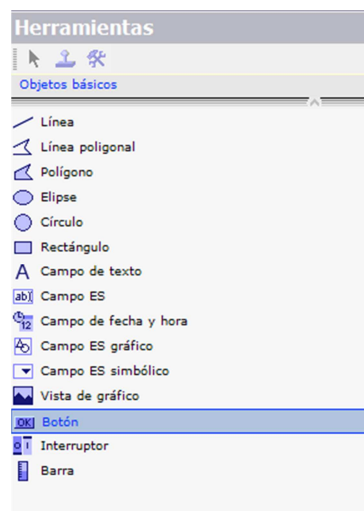


Figura 28: Botones

Podremos editar sus propiedades pinchando sobre el botón y accediendo a la ventana de propiedades del objeto en la parte inferior de la pantalla.

### Barras.

Las hemos utilizado para configurar el nivel de los depósitos. Ya que son indicadores de nivel, van variando según aumente o disminuya el mismo. Se encuentran en los *Objetos básicos* de las *Herramientas* como ya hemos visto con anterioridad con los botones o los campos de entrada/salida.

Si pinchamos sobre una barra, podremos configurarla desde la ventana de propiedades del objeto (*Figura 29*):

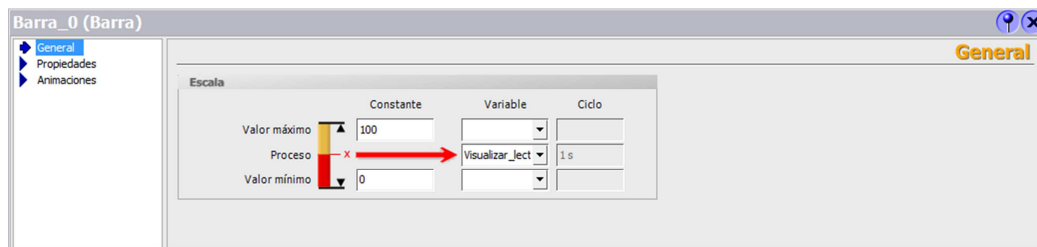


Figura 29: Propiedades de la barra

En el apartado *General* de las propiedades hemos asociado una variable a la barra. El valor de la misma irá variando según varíe el valor de la variable asociada. Por otro lado, en *Propiedades – Apariencia*, hemos cambiado su aspecto de forma que se pueda distinguir a simple vista, el nivel de qué depósito están representando tal y como vemos en la *Figura 30*:

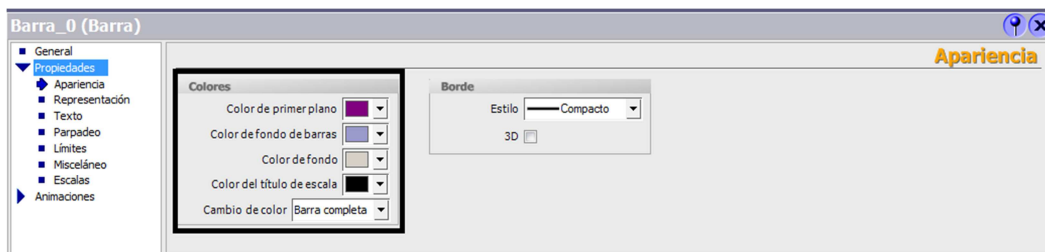


Figura 30: Apariencia de la barra

Por otro lado, en *Propiedades – Escalas*, hemos eliminado cualquier tipo de escala para darle un aspecto lo más parecido a los depósitos del sistema físico (*Figura 31*):

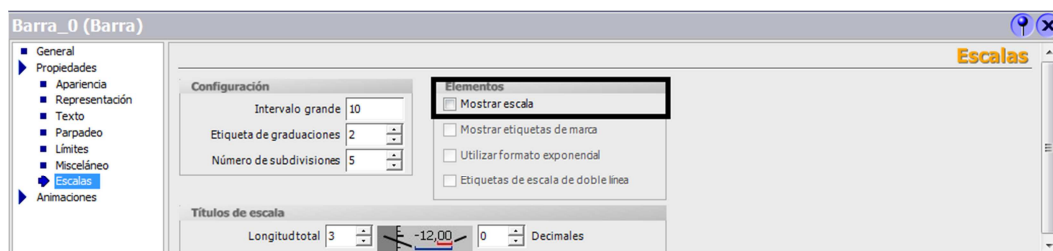


Figura 31: Quitar escalas



## Gráficos.

El resto de elementos corresponde a gráficos o representaciones gráficas (tuberías, válvulas, bombas...) obtenidas de las librerías anteriormente cargadas. Como ocurre con el resto de elementos explicados, si pinchamos sobre cualquier gráfico, accederemos directamente a sus propiedades y podrán ser modificadas según nuestras necesidades.

Ahora ya sabemos dónde tenemos que buscar y cómo configurar los distintos elementos que formarán las interfaces de nuestro sistema. Como vemos, es muy intuitivo elaborar imágenes o interfaces que conformarán nuestro sistema de control.

## 7.5. Lectura del sensor inteligente SITRANS Probe LU en WinCC flexible.

Mención aparte requiere la lectura del sensor en WinCC flexible. Para visualizar el valor del sensor en nuestra interfaz, vamos a necesitar de un campo de entrada/salida. Como ya hemos visto, estos campos se utilizan para representar valores numéricos de las variables. Crearemos una variable nueva y la asociaremos a la variable del autómeta, *valor\_sensor*. A continuación, en las propiedades del campo E/S, ajustaremos la salida de la forma: (Figura 32):

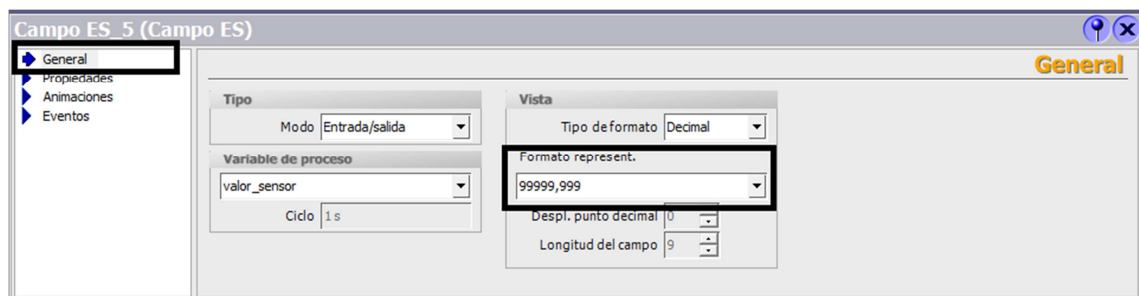


Figura 32: Representación del campo E/S

Hay que tener en cuenta que cuando configuremos la variable que mostrará la lectura del sensor, ésta debe tener el mismo tipo de datos que su homóloga en *Step 7*. No obstante, el sensor está midiendo distancia, por lo que un valor alto de lectura, gráficamente en *WinCC* será representado como un depósito muy lleno. Por lo tanto es necesario restar el valor que obtenga el sensor a la distancia máxima de la base del sensor al fondo del depósito. Esto debemos hacerlo en *Step 7*, en la interfaz de programación, en el *OBI*. Por tanto, configuraremos una nueva variable que llamaremos *nivel\_sensor* para obtener el resultado de la resta. La configuración en *OBI* se muestra en la Figura 33:

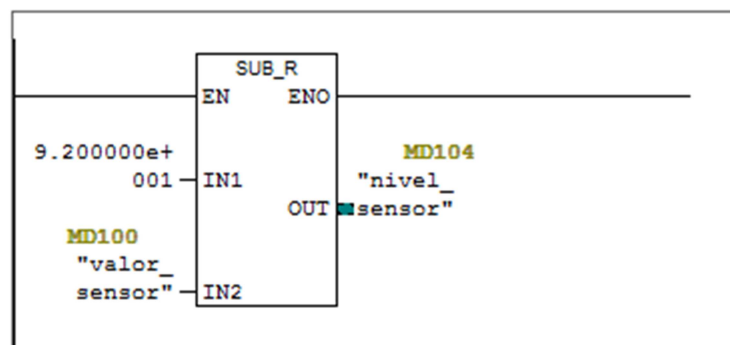


Figura 33: Obtención del nivel

Recordemos que estamos trabajando con números reales, por lo tanto, la función resta ha de ser con números reales. Una vez hecha la resta, asociaremos la variable al campo de *E/S* correspondiente y obtendremos el valor del nivel en la interfaz de simulación (*Figura 34*).

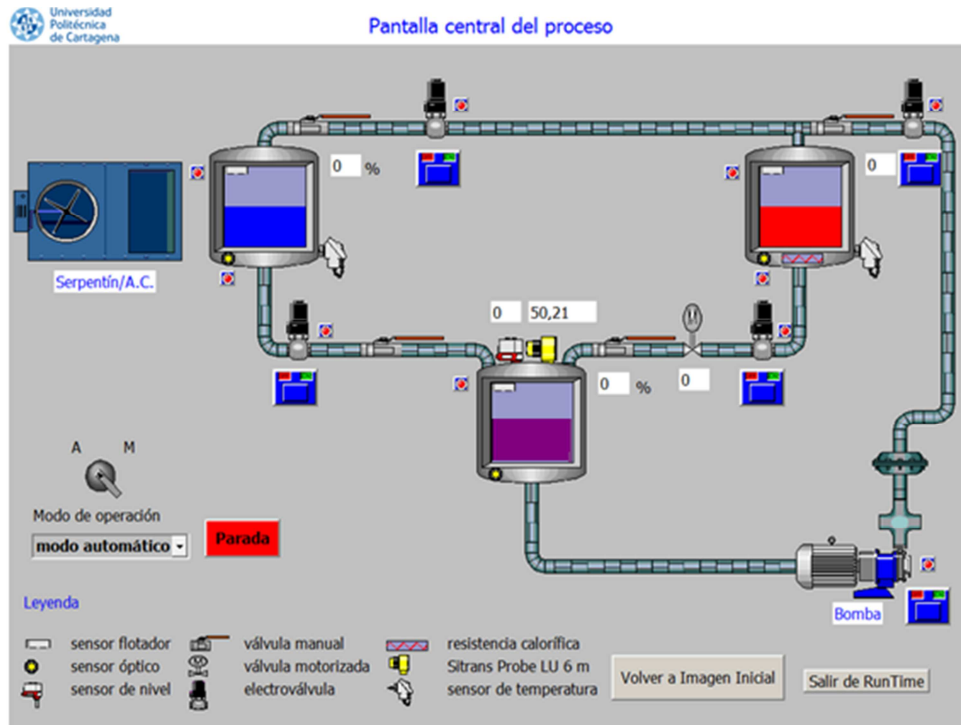


Figura 34: Interfaz de simulación

## 7.6. Navegación de imágenes y compilación del trabajo.

El último paso en *WinCC flexible* será configurar la navegación de las imágenes y compilar todo el trabajo. En la navegación de imágenes configuraremos la jerarquía de éstas, pudiendo pasar de una a otra. Para acceder a la configuración, vamos a la *Vista de Proyecto*, seleccionamos *Configuración del panel de operador* y pinchamos sobre *Navegación de imágenes*. Al hacer esto, aparecerá la siguiente imagen (*Figura 35*):

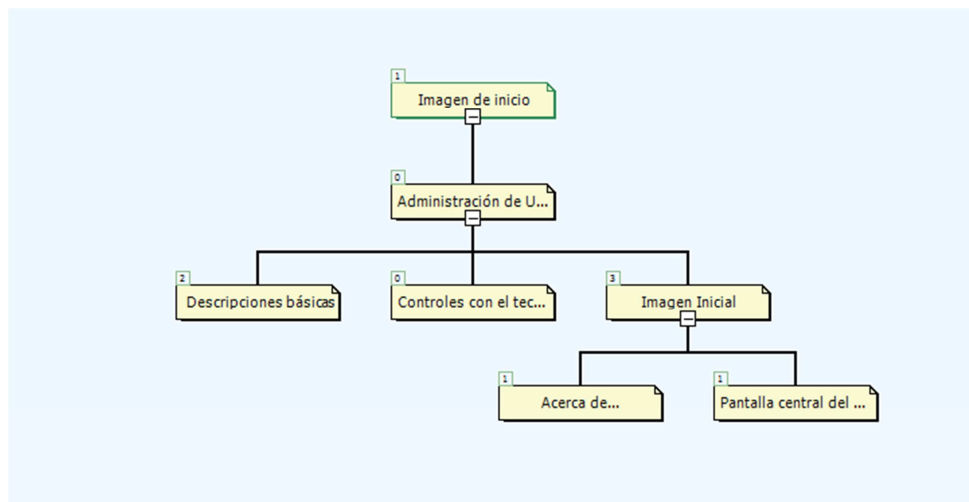


Figura 35: Jerarquía de imágenes

La navegación de imágenes de nuestro proyecto está definida según la imagen anterior. La configuración es sencilla. Simplemente tenemos que pinchar sobre el nombre de las imágenes (éstas aparecen automáticamente) y arrastrarlas según queramos movernos entre ellas.

Por último, compilaremos el programa. Para ello presionaremos el botón Iniciar *RunTime* de la barra de herramientas, tal y como muestra la siguiente imagen (*Figura 36*):

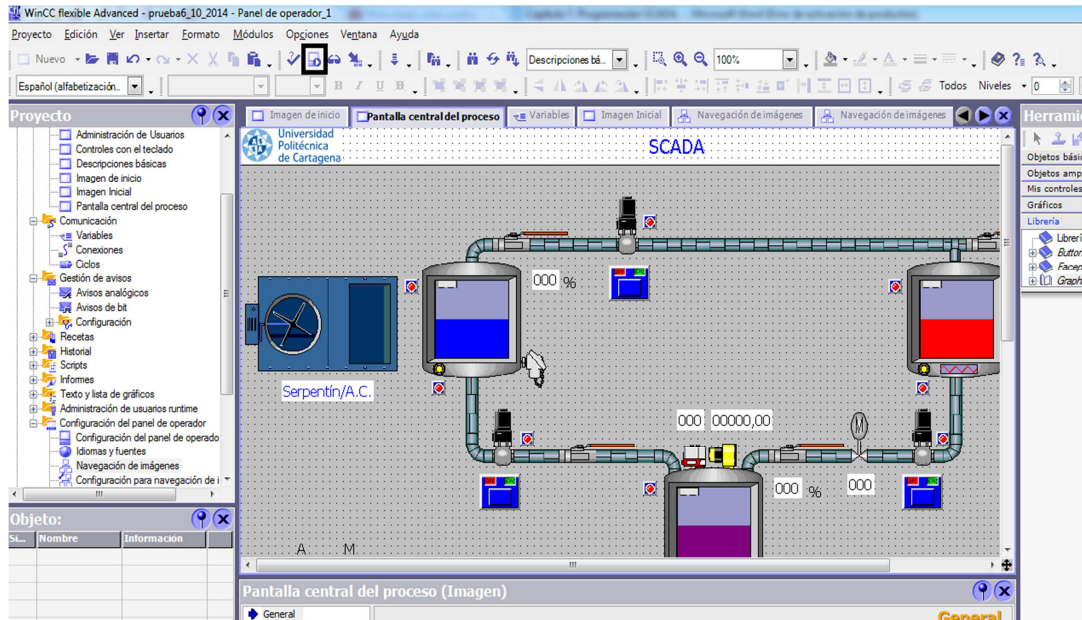


Figura 36: Iniciar *RunTime*

Si no hemos cometido fallos, el programa de simulación *RunTime* se iniciará sin problemas, apareciendo la siguiente *Figura 37*:



Figura 37: *RunTime* en marcha

## Capítulo 8: Conclusiones y trabajos futuros.

---

Cada vez más en la industria se hace necesario dominar y conocer las herramientas de programación y automatización. Así como las propias de visualización de procesos. En este trabajo hemos podido desarrollar una pequeña aplicación de cada una de estas herramientas, lo cual nos ha dotado de una percepción más amplia del entorno industrial.

El trabajo realizado es una pequeña gota en el océano de la automatización industrial, pero nos ha sido de gran ayuda como guía para adentrarnos en ese gran mundo.

Por otro lado, también hemos realizado un trabajo grande de calibración y parametrización del sensor inteligente *SITRANS Probe LU*. Esta fase del proyecto fue realmente laboriosa ya que carecíamos de los manuales adecuados y las herramientas necesarias. A base de probar, método del ensayo y error, hemos sido capaces de crear una guía de parámetros del sensor, de modo que los futuros alumnos que trabajen con este dispositivo, no empiecen desde cero.

Las dificultades a lo largo de este trabajo no han sido pocas. Aún así ha merecido la pena enfrentarse a ellas y superarlas con mayor o menor éxito; pero superarlas al fin. Todo ello ha sido una gran lección de aprendizaje y contacto, una vez más, con el mundo de la automatización industrial.

Este proyecto pretende ser eso, una guía práctica y docente que sirva de primer contacto con el entorno industria de automatización para los futuros alumnos. Una gran toma de contacto, pero pequeña comparada con el mundo industrial en sí.

Por su parte, el trabajo realizado aún puede ser ampliado. Y por ello, ya se está trabajando en futuros procesos de control con otro tipo de instrumentos adecuados para tal labor. De todos ellos el más llamativo, desde un punto de vista docente, sea la implementación de un control PID, el cual ampliará el trabajo realizado en este proyecto y dotará a los alumnos de una herramienta más a la hora de afrontar los distintos retos que les surgirán en la industria.

En conclusión, podemos decir que a pesar de lo laborioso del trabajo realizado, de las dificultades surgidas a lo largo del mismo y los quebraderos de cabeza, concluir este proyecto ha sido una herramienta muy útil de aprendizaje, ya que hemos podido tomar nociones de cada uno de los campos en los que se mueve la automatización: calibración de instrumentos, comunicación, programación y visualización de procesos. Todo ello nos ha dotado de unos recursos muy valiosos para adentrarnos en la automatización.

Al final hemos intentado cumplir, de la mejor manera posible, todos los objetivos marcados.

- Calibración de sensor inteligente
- Comunicación del mismo con el resto de hardware del proceso
- Programación del autómatas de control
- Visualización de todo el proceso

Siempre se podrá mejorar el trabajo realizado, pero tenemos la sensación de haber hecho algo bonito, motivador y útil. Y si puede servir como guía, la satisfacción es doble.